Oracle® Communications

Cloud Native Core, Converged Policy User Guide

Release 24.2.8
F83322-12
November 2025

Oracle Communications Cloud Native Core, Converged Policy User Guide, Release 24.2.8

F83322-12

Contents

8 Alerts

9 CNC Policy Metrics

10 CNC Policy KPIs

My Oracle Support

My Oracle Support (https://support.oracle.com) is your initial point of contact for all product support and training needs. A representative at Customer Access Support can assist you with My Oracle Support registration.

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at http://www.oracle.com/us/support/contact/index.html. When calling, make the selections in the sequence shown in the following list on the Support telephone menu:

- For Technical issues such as creating a new Service Request (SR), select **1**.
- For Non-technical issues such as registration or assistance with My Oracle Support, select **2**.
- For Hardware, Networking and Solaris Operating System Support, select **3**.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

What's New in This Guide

This section introduces the documentation updates for release 24.2.x.

Release 24.2.8 - F83322-12, November 2025

- Modified the placement of `USER.ssv.excludeDnn.<Snssai>` advanced settings key in the PCF Session Management.
- Modified the placement of `USER.ssv.excludeApns` advanced settings key in theSettings.

Release 24.2.8 - F83322-11, November 2025

- Added `USER.ssv.excludeDnn.<Snssai>` advanced settings key to PCF Session Management.
- Added `USER.ssv.excludeApns` advanced settings key to Settings.

Release 24.2.7 - F83322-10, July 2025

There are no changes to this document in this release.

Release 24.2.6 - F83322-09, June 2025

There are no changes to this document in this release.

Release 24.2.5 - F83322-08, April 2025

- Updated Error Code Dictionary with the error code dictionaries for Egress Gateway, Ingress Gateway, and NRF Client.

Release 24.2.4 - F83322-06, March 2025

- Added the Traffic Segregation section to support Ingress and Egress Gateway traffic segregation in Policy.
- **Message Feed for SBI Monitoring**

  Added/updated the following sections to include details of Policy Message Feed feature:
  - Added Message Feed for SBI Monitoring section to describe the Policy Message Feed feature.
  - Added details of the following metrics to Ingress Gateway Metrics:
    * oc_ingressgateway_msgcopy_requests_total
    * oc_ingressgateway_msgcopy_responses_total
  - Added details of the following metrics to Egress Gateway Metrics:
    * oc_egressgateway_msgcopy_requests_total
    * oc_egressgateway_msgcopy_responses_total
  - Added details of the following alerts:
    * INGRESS_GATEWAY_DD_UNREACHABLE_MAJOR
    * EGRESS_GATEWAY_DD_UNREACHABLE_MAJOR
- The following metrics in Pod Congestion Metrics section supports prefix and suffix:
  - `pod_cong_state_report`

  - pod_resource_congestion_state_report
- Added the following Advanced keys to PCRF Core Settings for handling the race condition between Gx and Sy over two sites for CCR-U.
  - DIAMETER.Gx.Update.RaceModeratorEnabled
  - DIAMETER.Gx.Update.RetryOnRarRaceEventTriggers
  - DIAMETER.Gx.Update.RetryAttemptsOnRarRace
  - DIAMETER.Gx.Update.RetryWaitTimeOnRarRace
  - DIAMETER.Gx.Update.RejectionErrorCodeOnRarRace
- Added the following Advanced keys to PCRF Core Settings for handling the race condition between Gx and Sy over two sites for CCR-T.
  - DIAMETER.Gx.Terminate.RaceModeratorEnabled
  - DIAMETER.Gx.Terminate.RetryOnRarRaceEventTriggers
  - DIAMETER.Gx.Terminate.RetryAttemptsOnRarRace
  - DIAMETER.Gx.Terminate.RetryWaitTimeOnRarRace
  - DIAMETER.Gx.Terminate.RejectionErrorCodeOnRarRace

Release 24.2.3 - F83322-05, January 2025

There are no changes to this document in this release.

Release 24.2.2 - F83322-04, November 2024

There are no changes to this document in this release.

Release 24.2.1 - F83322-02, October 2024

- Added or updated the following sections to add details on Concurrency Handling at Bulwark Service to Reduce Processing Latency of Service Request feature:
  - Added Concurrency Handling at Bulwark Service to Reduce Processing Latency of Service to describe the feature.
  - Added Bulwark with the configuration details of Bulwark service.
  - Added details of CONCURRENCY.LOCK_RETRY_MODE Advanced Settings key in:
    - PCF Session Management
    - PDS Settings
  - Added the following metrics to Bulwark Metrics:
    - lock_response_total
    - lock_response_total
    - lock_collision_total
  - Added the following alerts to List of Alerts:
    - LOCK_ACQUISITION_EXCEEDS_MINOR_THRESHOLD
    - LOCK_ACQUISITION_EXCEEDS_MAJOR_THRESHOLD
    - LOCK_ACQUISITION_EXCEEDS_CRITICAL_THRESHOLD
- Added the following sections to support prevention of excessive accumulation of requests at Undertow Worker Queue:

  - Added the Support for Prevention of Requests Accumulation at Undertow Worker Queue section to describe the functionality.
  - Added the following Undertow Worker Queue metrics to Undertow Server Metrics section:
    * `occnp_undertow_queue_limiter_reject_request_total`
    * `occnp_undertow_queue_limiter_accept_request_total`
    * `occnp_undertow_queue_request_limiter_active_threads_count`
- Updated the following sections for Pod Congestion Control feature to SM Service:
  - Updated the SM Service Pod Congestion Control section with CPU, Queue, and Discard Priority default values.
  - Updated the Thresholds section with threshold default values for SM service.
  - Updated the Load Shedding Rules section with discard priority default values for SM service.
- Updated the following sections for Pod Congestion Control feature to Bulwark Service:
  - Updated the Bulwark Pod Congestion Control section with configurations for enabling Congestion Control for Bulwark service.
  - Updated the Settings menu with configurations for enabling Congestion Control for Bulwark service.
- Added or updated the details of the following metrics in :
  - ue_n1_transfer_request_total
  - ue_n1_transfer_response_total
  - ue_n1_subscribe_request_total
  - ue_n1_subscribe_response_total

Release 24.2.0 - F83322-02, October 2024

- Added/updated the following metrics in PerfInfo Metrics:
  - load_level
  - service_resource_stress
  - service_resource_overload_level
  - system_overload_threshold_config_mode
  - active_overload_threshold_fetch_failed
  - load_level_report_total
  - service_resource_overload_level_report_total
  - http_out_conn_request
  - http_out_conn_response
  - overload_manager_enabled
  - leader_pod
- Updated Support for End-to-End Log Identifier Across Policy Services with the details of the callflows and services for which end-to-end log identifier is not yet supported.

Release 24.2.0 - F83322-03, October 2024

- Added/updated the following metrics in PerfInfo Metrics:
  - load_level
  - service_resource_stress
  - service_resource_overload_level
  - system_overload_threshold_config_mode
  - active_overload_threshold_fetch_failed
  - load_level_report_total
  - service_resource_overload_level_report_total
  - http_out_conn_request
  - http_out_conn_response
  - overload_manager_enabled
  - leader_pod
- Updated Support for End-to-End Log Identifier Across Policy Services with the details of the callflows and services for which end-to-end log identifier is not yet supported.

Release 24.2.0 - F83322-01, August 2024

- Added details of AUDIT.HTTP2_ENABLED advanced settings key to PCF Session Management
- Updated Support for Adding Reduced Capability to UEs section to mention the Adding Reduced Capability to UEs feature support for AM service and UE Policy service.
- Updated Support for End-to-End Log Identifier Across Policy Services section to mention Support for End-to-End Log Identifier Across Policy Services feature support for all Policy microservices.
- Updated the Support for Concurrency Handling using Bulwark Service in PCRF section with details of bulwark service in PCRF for Rx messages.
- Added the Pending Transactions on Gx Interface section to describe the pending transaction on Gx interface.
- Added the following sections to Support for policyDecFailureReports Attribute feature:
  - Added Support for policyDecFailureReports Attribute section to describe the enhancements to error response feature.
  - Added `RULE.ENABLE_PCC_RULE_REMOVE_ON_FAILURE` key in the Advanced Settings of PCF Session Management.
  - Added *occnp_feature_info_received_total* metric in SM Service Metrics:
- Added the following sections to Enhancements to Error Response feature:
  - Added Enhancements to Error Response section to describe the enhancements to error response feature.
  - Added Error Code Dictionary section to with details of the error code dictionaries for AM, UE, SM, UDR, CHF, and Binding services.
  - Updated Error Handling section with details of configuring error handling functionality for AM, UE, SM, UDR, CHF, and Binding services.

  - Added details of the following metrics in CNC Policy Metrics:
    - error_handler_exec_total
    - error_handler_in_total
    - error_handler_out_total
- Added the following sections to describe Optimizing Database Encoding in PCRF Core service:
  - Added Support for Optimizing Database Encoding in PCRF Core section to describe the database encoding in PCRF Core services.
  - Added details of the following advanced settings keys to PCRF Core Settings section.
    - `DB.GX.DATA.ENCODING.Enabled`
    - `DB.RX.DATA.ENCODING.Enabled`
    - `DB.SD.DATA.ENCODING.Enabled`
    - `DB.GX.ENCODING.MAP.Version`
    - `DB.RX.ENCODING.MAP.Version`
    - `DB.SD.ENCODING.MAP.Version`
    - `DB.ENCODING.MAP.LIST`
  - Added the following metrics to PCRF Core Metrics settings.
    - `occnp_data_encoding_total`
    - `occnp_data_decoding_total`
    - `occnp_data_decoding_fail_total`
    - `occnp_data_encoding_size_before_total`
    - `occnp_data_encoding_size_after_total`
- Added the following sections to describe Congestion Control feature in PDS service:
  - Added PDS Pod Congestion Control section to describe the pod congestion control mechanism supported by PDS service.
  - Updated the Congestion Control section in CNC Console to support congestion control for PDS service.
  - Added details of the following advanced settings keys to PDS Settings section:
    - `PDS_NOTIFY_USER_DATA_REQUEST_PRIORITY`
    - `PDS_GET_DEFAULT_WORKFLOW_REQUEST_PRIORITY`
    - `PDS_GET_USER_DATA_REQUEST_PRIORITY`
    - `PDS_UPDATE_USER_DATA_REQUEST_PRIORITY`
    - `PDS_DELETE_USER_DATA_REQUEST_PRIORITY`
    - `PDS_AUDIT_NOTIFY_REQUEST_PRIORITY`
    - `CONGESTION_RESPONSE_CODE`
  - Added the following alerts in the Common Alerts section:
    - POD_CONGESTION_L1
    - POD_CONGESTION_L2

        * POD_PENDING_REQUEST_CONGESTION_L1
        * POD_PENDING_REQUEST_CONGESTION_L2
        * POD_CPU_CONGESTION_L1
        * POD_CPU_CONGESTION_L2
- Added the following sections to describe Congestion Control feature in Usage Monitoring service:
  - Added Usage Monitoring Pod Congestion Control section to describe the pod congestion control mechanism supported by Usage Monitoring service.
  - Updated the Congestion Control section in CNC Console to support congestion control for Usage Monitoring service.
  - Added details of the following fields to Configuring Usage Monitoring section to support congestion control for Usage Monitoring service:
    * `UM Session Create API`
    * `UM Session Update API`
    * `UM Session Terminate API`
    * `UM Session Notify API`
    * `UM Session Audit Subscriber API`
    * `UM Session Search Subscriber API`
    * `UM Session Audit Notify API`
    * `Congestion Error Code`
  - Added the metric `um_http_congestion_message_reject_total` to the Pod Congestion Metrics section.
- Updated the following sections to Sy SLR Enhancements for Signalling Updates and UDR Notification feature:
  - Added Handling N28 and N36 Interfaces Context Information during Subscription Failures section describing the PDS service storing the context information to the database.
  - Added details of the following advanced settings keys to PCRF Core service section:
    * `USER.smPolicyData.createContextOnFailure`
    * `USER.operatorSpecificData.createContextOnFailure`
  - Added details of the following advanced settings keys to SM Service section:
    * `USER.CREATE_CONTEXT_ON_FAILURE_SM_POLICY_DATA`
    * `USER.CREATE_CONTEXT_ON_FAILURE_CHF_DATA`
    * `USER.CREATE_CONTEXT_ON_FAILURE_OPERATOR_SPECIFIC_DATA`
- Added the following sections to describe Congestion Control feature in SM Service:
  - Added SM Service Pod Congestion Control section to describe the pod congestion control mechanism supported by SM service.
  - Updated the Congestion Control section in CNC Console to support congestion control for SM service.
  - Added details of the following advanced settings keys to PCF Session Management section:

    - SM.UPDATE.EVENT.SUBS.PRIORITY
    - SM.CREATE.PRIORITY
    - SM.SUB.FAIL.NOTIFY.PRIORITY
    - SM.USER.SERVICE.NOTIFY.PRIORITY
    - SM.UPDATE_PRIORITY
    - SM.REAUTH_PRIORITY
    - SM.DELETE.PRIORITY
    - SM.POLICY.CLEANUP.PRIORITY
    - SM.APP.SESSION.CREATE.PRIORITY
    - SM.APP.SESSION.CLEANUP.PRIORITY
    - SM.AUDIT.NOTIFY.PRIORITY
    - SM.GET.APP.SESSION.PRIORITY
    - SM.GET.ASSOC.PRIORITY
    - SM.GET.SUBSCRIBER.SESSIONS.PRIORITY
    - SM.GET.ASSOC.QUERY.PRIORITY
    - CONGESTION_RESPONSE_CODE
- Added *Limiting the number of sessions per subscriber and DNN+SNSSAI for SM service and PCRF Core service* section to Limiting the Number of Sessions to describe limiting the number of sessions for SM sessions and PCRF Core sessions.
- Updated the following sections to describe GxSession database slicing for PCRF Core:
  - Added *Slicing in GxSession database for PCRF Core service* to Support for Database Slicing to describe GxSession database slicing for PCRF Core service.
  - Added details of `DISTRIBUTE_GX_TRAFFIC_USING_TABLE_SLICING` advanced settings key to Settings, which is used to distribute the traffic for GxSession database across all the slices.

Acronyms

The following table lists the acronyms and the terminologies used in the document:

Table Acronyms and Terminologies

Acronym	Definition
3GPP	3rd Generation Partnership Project
AAA	Authorization Authentication Answer
AAR	Authorization Authentication Request
AF	Application Function
AMF	Access and Mobility Management Function
API	Application Programming Interface
ARS	Alternate Route Selection
ASM	Aspen Service Mesh
ASR	Abort-Session-Request
ATS	The core service sends the subscriber state variables to PDS only when there is an update to the variables.
AVP	Attribute Value Pair
BSF	Oracle Communications Cloud Native Core, Binding Support Function
CA	Certificate Authority
CDCS	Oracle Communications CD Control Server
CHF	Charging Function
CM	Configuration Management
CNC	Cloud Native Core
CNC Console	Oracle Communications Cloud Native Configuration Console
CNE	Oracle Communication Cloud Native Core, Cloud Native Environment
CNPCRF	Oracle Communications Cloud Native Core, Policy and Charging Rules Function
CUSTOMER_REPO	Docker registry address including the port number, if the docker registry has an associated port.
cnDBTier	Oracle Communications Cloud Native Core, cnDBTier
DNS	Domain Name System
DRA	Diameter Routing Agent
FQDN	Fully Qualified Domain Name
GUAMI	Globally Unique AMF Identifier
IMAGE_TAG	Image tag from release tar file. You can use any tag number. However, make sure that you use that specific tag number while pushing docker image to the docker registry.
IMS	IP Multimedia Subsystem
HTTPS	Hypertext Transfer Protocol Secure
MCC	Mobile Country Code
MCPTT	Mission-critical push-to-talk
METALLB_ADDRESS_POOL	Address pool configured on metallb to provide external IPs
MNC	Mobile Network Code
NEF	Oracle Communications Cloud Native Core, Network Exposure Function

Table (Cont.) Acronyms and Terminologies

Acronym	Definition
NF	Network Function
NPLI	Network Provided Location Information
NRF	Oracle Communications Cloud Native Core, Network Repository Function
OSO	Oracle Communications Operations Services Overlay
P-CSCF	Proxy Call Session Control Function
PA Service	Policy Authorization Service
PCC	Policy and Charging Control
PDB	Pod Disruption Budget
PLMN	Public Land Mobile Network
PCF	Oracle Communications Cloud Native Core, Policy Control Function
PCRF	Oracle Communications Cloud Native Core, Policy and Charging Rules Function
PCEF	Policy and Charging Enforcement Function
PCSCF	Proxy Call Session Control Function
PDS	Policy Data Service
PRA	Presence Reporting Area
PRE	Policy Runtime Engine
PDU	Protocol Data Unit
Policy	Oracle Communications Cloud Native Core, Converged Policy
QoS	Quality of Service
RAA	Re-Auth-Answer
RAN	Radio Access Network
RAR	Re-Auth-Request
SBI	Service Based Interface
SAN	Subject Alternate Name
SCP	Oracle Communications Cloud Native Core, Service Communication Proxy
SMF	Session Management Function
S-NSSAI	Single Network Slice Selection Assistance Information
UDR	Oracle Communications Cloud Native Core, Unified Data Repository
SRA	Successful Resource Allocation
STR	Session Termination Request
TTL	Time To Live
UE	User Equipment
UPF	User Plane Function
UPSI	UE Policy Section Identifier
URSP	UE Route Selection Policies
UPSC	UE Policy Section Code
URI	Uniform Resource Identifier
VSA	Vendor Specific Attributes

1 Introduction

This document provides information about the role of Oracle Communications Cloud Native Core, Converged Policy (Policy) in 5G Service Based Architecture and how to configure and use Policy services and managed objects.

1.1 Overview

Policy is a key component of the 5G Service Based Architecture(SBA). It provides a flexible, secure, and scalable policy designing solution. Policy interacts with other network functions to perform usage monitoring, network behavior management, and governance. It helps operators to design, test, and deploy different network policies supporting 5G deployments. Policy is designed and built with a microservice based architecture on cloud native principles. It uses network, subscriber, and service information to help service providers create policies and determine how and under what conditions subscribers and applications use network resources. It helps in minimizing network utilization while maximizing the quality of experience for operators. Policy solution supports deployments into any cloud, including containers on Bare Metal managed by Kubernetes or VMs managed by OpenStack.

> **Note**
>
> The performance and capacity of the Policy system may vary based on the Call model, Feature/Interface configuration, underlying CNE and hardware environment, including but not limited to the complexity of deployed policies, policy table size , object expression and custom json usage in Policy design.

Policy is a network function for policy control decision and flow based charging control. It consists of the following functions:

- Policy rules for application and service data flow detection, gating, QoS, and flow based charging to the Session Management Function (SMF)
- Access and Mobility Management related policies to the Access and Mobility Management Function (AMF)
- UE Route Selection Policies (URSP) rules to User Equipement (UE) through AMF
- Access to subscription information relevant for policy decisions in a Unified Data Repository (UDR)
- Network control for service data flow detection, gating, and Quality of Service (QoS)
- Flow based charging towards the Policy and Charging Enforcement Function (PCEF)
- Receiving session and media related information from Application Function (AF) and informing AF of traffic plane events
- Provision of Policy and Charging Control (PCC) Rules to Policy and Charging Enforcement Function (PCEF) through the Gx reference point

Policy supports the above functions through the following services:

- Session Management Service

- Access and Mobility Service
- Policy Authorization Service
- User Equipment (UE) Policy Service
- PCRF Core Service
- Binding Service
- Policy Data Source Service
- Usage Monitoring Service
- Notifier Service
- NWDAF Agent

1.2 References

You can refer to the following documents for more information.

- *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Converged Policy REST API Specification Document*
- *Oracle Communications Cloud Native Core, Converged Policy Design Guide*
- Visual Programming Editor: https://developers.google.com/blockly
- 3GPP Technical Specification 29.512 v15.3.0, Session Management Policy Control Service, Stage 3, Release 15
- 3GPP Technical Specification 29.514 v15.3.0, Policy Authorization Service, Stage 3, Release 15
- 3GPP Technical Specification 29.507 v15.3.0, Access and Mobility Policy Control Service, Stage 3, Release 15
- 3GPP Technical Specification 29.525 v15.5.1, UE Policy Control Service, Stage 3, Release 15
- 3GPP Technical Specification 29.518 v15.5.1, Access and Mobility Management Services, Stage 3, Release 15

2
CNC Policy Architecture

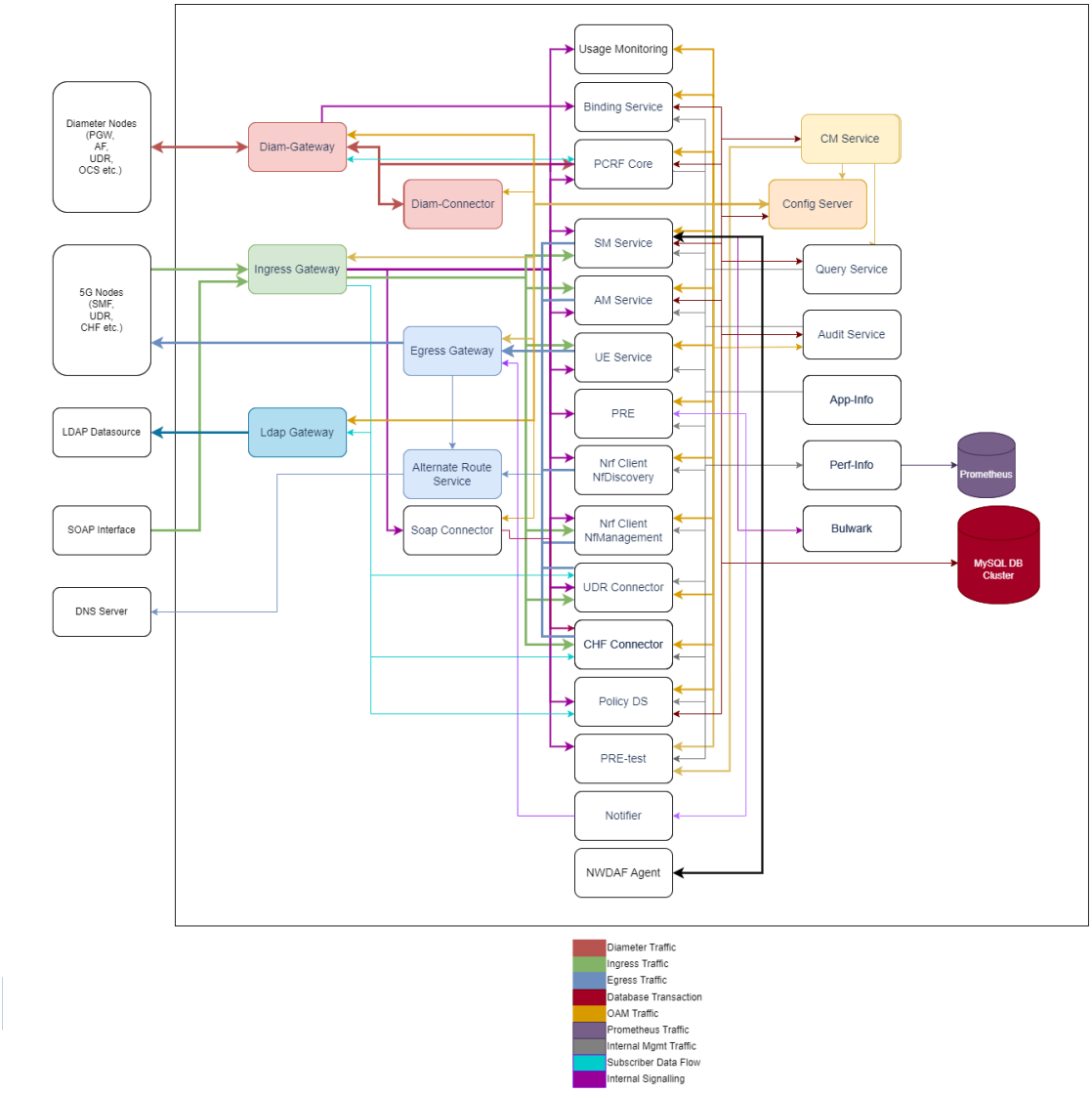
Oracle Communications Cloud Native Core, Converged Policy (Policy) is developed as a cloud native application that is composed of a collection of microservices that run in a cloud native environment. It separates the processing or business logic into the logical grouping of microservices and components:

- **Connectivity**: Components interfacing with external entities. This is where an API gateway is utilized to interface with external traffic to the PCF. These are stateless sets of components.
- **Business logic**: Application layer running the PCRF or PCF business logic, policy engine, and various services that can be enabled based on deployment requirements. These are stateless sets of components.
- **Data Management**: Data layer responsible for storing various types of persistent data. PCF is developed to be able to plug in different types of back-end data layers that could be internal or external.

The Policy solution provides a flexible and modular policy designing framework. It offers, rapid and secure deployment of new policies and supports the existing use cases. The Converged policy solution supports both 4G and 5G networks, thereby helping operators to manage their heterogeneous network in an intuitive and consistent manner while enabling seamless inter-working and migration between 4G and 5G networks.

The following diagram represents the Policy architecture:

Figure 2-1 Policy Architecture Diagram



Components of the Policy Architecture

- Kubernetes cluster hosting Docker containers and Calico networking
- Optional CNE services to support operation of PCF
- Connectivity
  - **Diameter Gateway/Connector** – Enables the policy solution functions as a Diameter server and offers integration over Gx, Rx, Sy and other legacy Diameter services. The Diameter server also implements routing, load balancing and overload control services. The Diameter Gateway acts as a gateway for all Diameter traffic to Policy Solution. It also performs round-robin load balancing across its back-end peers (Diameter connector and PCRF-Core).

> **Note**
>
> For the Diameter Gateway, minimum and maximum replica settings cannot be configured because it is stateful. For StatefulSets (STS), only the total number of replicas can be set.

  - **Ingress Gateway** – Acts as a Gateway for all ingress HTTP traffic to Policy Solution.
  - **Egress Gateway** – Acts as a gateway for all egress HTTP traffic originating from Policy Solution to outside the network.
  - **LDAP Gateway** – Acts as a gateway for all egress LDAP traffic towards Directory Services.
  - **Diameter Connector** – Accepts Diameter messages from Diameter Gateway and converts the message to HTTP message format and sends to PCF components.
  - **Soap Connector** – Accepts the SOAP messages from ingress gateway, converts to JSON format and forwards the message to Policy Data Service for processing.
  - **NRF Client Service** – Integrates with NRF for service registration, discovery, and service status or load related information, along with application and performance information services. NRF discovery helps in on-demand discovery of network functions. NRF management helps in autonomous discovery of network functions.
- Policy Business Logic
  - **SM Service (includes PA Service)** - Provides the SMF session and application or flow based policies. The Policy Authorization (PA) service, such as Rx like interface in SBA authorizes an AF request and creates policies as requested by the NF consumer service for the PDU session to which the AF session is bound. This service implements policy control for session management for service data flows. This service implements the N7 interface to trigger session management policies towards the SMF function.
  - **AM Service** - Implements access management service-related policies over the N15 interface towards the AMF.
  - **PCRF Core Service** – Implements the legacy handling of PCRF core business logic, interactions with other microservices, and triggers for policy enforcement over the Gx interface. PCRF is a node that determines policy rules in a multimedia network in real-time.
  - **Binding Service** - Stores binding information related to 4G/5G subscribers and helps Diameter Gateway in forwarding AF messages.
  - **UE Policy Service** - Provides UE policy, includes UE Route Selection Policy (URSP) through AMF transparently to the UE. Implements UE management service-related policies over the N15 interface towards the AMF.
  - **UDR Connector** – UDR Connector layer interfaces the application with the UDR.
  - **CHF Connector** – CHF Connector layer interfaces with the the CHF.
  - **Policy Data Service**– Policy DS interfaces 4G/5G Signaling components with Protocol specific connectors (UDR Connector/CHF Connector/LDAP Gateway) to have a unified data source layer.
  - **Policy Runtime Engine** – Policy Runtime Engine (PRE) service runs the Policy Decision Engine. The policies can be configured using the configuration management service.

  - **PRE Test Engine** – The PRE Test Engine runs the Policy Decision Engine for test messages. Test message can be triggered from the configuration management service.
  - **Configuration Management** – This service provides the OAM interfaces that includes GUI and REST interfaces, for Policy and Service provisioning. Configuration Service and CM GUI offers graphical interface for all policy-related configurations and design of policies.
  - **Configuration Server** – This service performs the database abstraction for storage and retrieval of policy configuration.
  - **Query** – The Query microservice processes session viewer queries triggered by the configuration management service.
  - **Audit** – The Audit microservice runs the Audit engine to detect and process stale session records.
  - **App-Info** – This microservice monitors application (microservice) health and status.
  - **Perf-Info** – This microservice monitors application (microservice) capacity and load status.
  - **Bulwark** - This microservice facilitates the concurrency support for other internal services, such as SM service, AM service, Policy DS, etc.
  - **Notifier** - This microservice notifies subscribers about their data usage at different threshold levels.
  - **Usage Monitoring** - This microservice implements the usage monitoring procedures like usage accumulation, grant calculation, etc. for session and PCC flows.
  - **NWDAF Agent** - This microservice inetegrates Policy with Network Data Analytics Function (NWDAF) service to get analytics information.
- Data Tier
  - **Dynamic state** – Stores the session information relevant for policy context.
  - **Configuration store** – Stores the configuration related data.
- Ingress and Egress Gateway Traffic Management
  For more information on Ingress and Egress Gateway Traffic Management, see *Oracle Communications Cloud Native Core, Cloud Native Environment User Guide.*

3
Policy Services

This chapter explains the Policy services.

> **Note**
>
> The performance and capacity of the Policy system may vary based on the Call model, Feature/Interface configuration, underlying CNE and hardware environment, including but not limited to the complexity of deployed policies, policy table size , object expression and custom json usage in Policy design.

3.1 Audit Service

Policy signaling services like SM service, AM service, UE service, Binding Management service etc are stateless, thereby making the centralized DB tier to store the session states. The policy micorservices session processing and DB tier transactions happen over network. The transactions that fail due to varied reasons get stored in the database as stale records. These stale records grows indefinitely over a period of time. The Audit Service is responsible for auditing the database to monitor for stale records, and either notify or clean up these stale records. The service that needs to use the Audit service to audit its tables, must register with the Audit service.

Figure 3-1 Audit Service Registration

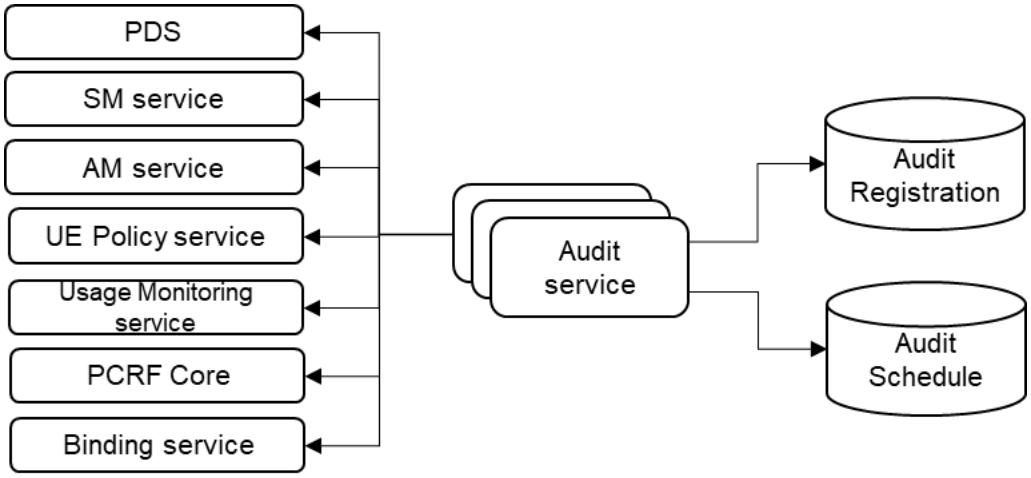
Audit service is mainly responsible for:

- allows core services to dynamically register or de-register the tables for auditing process
- core services can change other attributes at run time by using the Audit service restful interface
- notifying the context owners about the expired records in the registered table for auditing
- provide flow control of notifications towards the context owners
- perform forceful deletion of stale records, in case no corrective action is taken by context owners
- publish total count of records in the registered table for auditing

Audit service is not only responsible for stale session cleanup but has increased scope in other signaling features as well e.g., notifying core service for binding retry attempts.

Audit service with single pod deployment makes it neither scalable nor highly available. Audit service supports multiple pods, by using Audit Schedule. Audit Schedule helps to effectively manage and plan the audit service on the registered services.

Figure 3-2 Multiple Pod Audit Service



Exception tables for the following services are maintained in an NDB Cluster to capture conflicting data.

- Binding Service
- SM Service
- AM Service
- UE Policy Service
- PCRF Core
- PDS
- Usage Monitoring

These exception tables are used to detect occurrences of conflict between sites and records/keys in tables for which conflict happens.

Records in exception tables can grow rapidly due to frequent collision that can happen. Audit service is used to cleanup these exception tables in scheduled manner. The Audit service configuration for cleaning up these exception tables are predefined using Helm parameters at the time of installation with a default 24hr frequency for auditing.

To enable auditing on the exception tables, add `exceptionTableAuditEnabled` parameter to custom-values.yaml file while upgrading to latest version and set the value of this parameter to true. After the upgrade procedure is complete, enable or disable the audit for the above mentioned services using CNC Console. For more details on enabling the audit, see *Enabling/Disabling Services Configurations* section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.

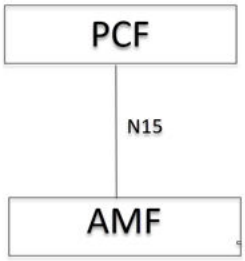
For information on how to configure Policy Audit Service, see Audit Service

3.2 Access and Mobility Service

Policy implements Access and Mobility (AM) management service related policies over the N15 interface towards Access and Mobility Management Function (AMF).

The following figure describes the communication between PCF and AMF over the N15 interface:

Figure 3-3 Communication between PCF and AMF over N15



AMF can perform the following functions:

- Enforcing control of policy decisions related to Radio Access Technology (RAT) or Frequency Selection Priority.
- Enforcing Service Area Restrictions. It is executed in User Equipement (UE).
- Enabling location tracking for a UE to get periodic updates on the current location of a subscriber.

Policy supports the following 3GPP defined services for AM Management:

Table 3-1 Access and Mobility Services

Service Operation Name	Description	Initiated By	Resource URI	HTTP Method
Npcf_AMPolicyControl_Create	Creates an AM Policy Association and provides corresponding policies to the Network Function (NF) consumer	AMF	{apiRoot}/npcf-am-policy-control/v1/policies/	POST
Npcf_AMPolicyControl_Update	Updates an AM Policy Association and provides corresponding policies to the NF consumer when the policy control request trigger is met or the AMF is relocated due to the UE mobility and the old PCF is selected	AMF	{apiRoot}/npcf-am-policy-control/v1/policies/{polAssoId}/update	POST
Npcf_AMPolicyControl_UpdateNotify	Provides updated policies to the NF consumer	PCF	{{Notification URI}/update {Notification URI}/terminate	POST

Table 3-1 (Cont.) Access and Mobility Services

Service Operation Name	Description	Initiated By	Resource URI	HTTP Method
Npcf_AMPolicyControl_Delete	Provides means for the NF consumer to delete the AM Policy Association	AMF	{apiRoot}/npcf-am-policy-control/v1/policies/{polAssoId}	DELETE
Npcf_AMPolicyControl_TerminateNotify	Requests termination of policies to the NF consumer	PCF	{apiRoot}/npcf-am-policy-control/v1/policies/{supi}/updateNotify	POST

For information on how to configure PCF Access and Mobility service, see PCF Access and Mobility.

3.3 Authorization Service

PCF implements policy authorization service that authorizes Application Function (AF) request over the N5 interface.

Policy Authorization service supports the creation of policies as requested by AF for Packet Data Unit (PDU) session. Policy authorization service is a critical function for IP Multimedia Subsystem (IMS) integration and dynamic Policy and Charging Control (PCC) rule creation.

Oracle Communications PCF supports the following 3GPP defined services for Policy Authorization:

Table 3-2 Policy Authorization Services

Service Operation Name	Description	Initiated By	Resource URI	HTTP Method
Npcf_PolicyAuthorization_Create	Determines and installs the policy according to the service information provided by an authorized NF service consumer.	AF, Network Exposure Function (NEF)	{apiRoot}/npcf-policyauthorization/v1/app-sessions	POST
Npcf_PolicyAuthorization_Update	Determines and updates the policy according to the modified service information provided by an authorized NF service consumer.	AF, NEF	{apiRoot}/npcf-policyauthorization/v1/app-sessions/{appSessionId}	PATCH
Npcf_PolicyAuthorization_Delete	Provides means to delete the application session context of the NF service consumer.	AF, NEF	{apiRoot}/npcf-policyauthorization/v1/app-sessions/{appSessionId}/delete	POST

Table 3-2 (Cont.) Policy Authorization Services

Service Operation Name	Description	Initiated By	Resource URI	HTTP Method
Npcf_PolicyAuthorization_Notify	Notifies NF service consumer of the subscribed events.	PCF	{notifUri}/notify {notifUri}/terminate	POST
Npcf_PolicyAuthorization_Subscribe	Allows NF service consumers to subscribe to the notification of events.	AF, NEF	{apiRoot}/npcf-policyauthorization/v1/app-sessions/{appSessionId}/events-subscription	PUT
Npcf_PolicyAuthorization_Unsubscribe	Allows NF service consumers to unsubscribe to the notification of events.	AF, NEF	{apiRoot}/npcf-policyauthorization/v1/app-sessions/{appSessionId}/events-subscription	DELETE

Policy authorization for PCF interaction with NEF

The `Npcf_PolicyAuthorization` on N5 interface acts as the main integration point between PCF and NEF. The `Npcf_PolicyAuthorization` ensures that the services and applications utilizing the 5G network, especially those accessed externally through the NEF, are governed by appropriate policies established by the PCF.

The `Npcf_PolicyAuthorization` service operation authorizes the request from the NF service consumer NEF, and optionally communicates with `Npcf_SMPolicyControl` service to determine and install the policy according to the information provided by the NF service consumer.

PCF complies with 3GPP 23.502 and 29.514 Rel 15.4. For details on the message format, procedures followed, and the other aspects of `Npcf_PolicyAuthorization` over N5 interface that PCF complies with, see the *Compliance Matrix*.

For configuring Policy Authorization service, see PCF Policy Authorization.

3.4 Binding Service

Binding service stores binding information related to 4G and 5G subscribers and help Diameter Gateway in forwarding Application Function (AF) requests. It also queries Oracle Communications Cloud Native Core, Network Repository Function (NRF) client for fetching Oracle Communications Cloud Native Core, Binding Support Function (BSF) information (One time Query and subscribe for notifications). For BSF, autonomous NRF discovery and static configuration is only supported, on-demand is not yet supported. Session Management service and PCRF- Core service send all the relevant information to Binding service asynchronously.

In Policy, Diameter Gateway should have a converged mode, where:

- Diameter Gateway connects to PCRF-Core
- Diameter Gateway also accepts a connection from Diameter Connector

In Converged mode of Diameter Gateway,

- All Diameter Applications except Rx are routed to PCRF-Core.

- For Rx messages, Diameter Gateway queries (asynchronously) Binding Service and based on the response, it forwards AF requests to either PCRF-Core or SM Service (via. Diameter Connector).

Configuration

In Helm Charts, you must configure **diamgateway.envGatewayMode** as **converged** in custom-values.yaml file, and the child helm files for Diameter-Gateway, Query-Service, and so on need to refer to this environment variable. This is mainly for performance considerations so that Diameter-Gateway and Query Service need not to query Binding service for the binding information. For more information on **diamgateway.envGatewayMode**, see *Oracle Communications Cloud Native Core Policy Installation, Upgrade and Fault Recovery Guide*.

Depending on the value of the following parameters, binding service or BSF is queried:

- **bindingEnable** in custom-values.yaml file
- **Binding Operation** in Oracle Communications Cloud Native Core, Cloud Native Configuration Console (CNC Console)

Behaviour is as follows:

- When both the configuration variables are set to true, binding service is queried.
- When the **bindingEnable** parameter is set to true and the **Binding Operation** is configured to false, BSF is queried.
- When the **bindingEnable** parameter is set to false and the **Binding Operation** is configured to true, neither BSF nor binding service is queried.
- When both the configuration variables are set to false, neither BSF nor binding service is queried.

Binding Service Truth table when receiving AAR-I

Following truth table is considered by Binding Service for finding the context owner of a session.

Priority: Priority indicates that attributes if specified in the message is considered as per the following priority from high to low. For example, IPv6 has higher priority indicates that if IPv6 is present, then only IPv6 binding is considered and corresponding context-owner information is returned otherwise next attribute priority is considered.

- IPv6
- IPv4 + APN/DNN+SNSSAI
- IPv4
- IMSI
- MSISDN

Table 3-3 Binding Service Truth table when receiving AAR-I

S.No.	Message	IPv6	IPv4	APN/ DNN+SN SSAI	IP Domain ID	GPSI/ MSISDN	SUPI/ IMSI	Binding Lookup Query
1	Gx-CCR I/SM Associati on	P (Present in Gx CCR-I/N7 Sm create message)	P	P	P	P	P	IPv6
	AAR - I	P (Present in Rx AAR-I message)	P	P	P			
2	Gx-CCRI/SM Associati on	P	P	P	P	P	P	IPv4 + APN/ (DNN+SN SSAI) + IP Domain ID
	AAR - I		P	P	P			
3	Gx-CCRI/SM Associati on	P	P	P	P	P	P	IPv4 (DNN and IP Domain ID won't be considere d)
	AAR - I		P			P	P	
4	Gx-CCRI/SM Associati on	P	P	P		P	P	IPv4 + APN/ (DNN+SN SSAI) + IP Domain IDIF No Records found (Would consider IP Domain ID as BLANK in Gx Session) => Not in the scope of PI-C (Need to enhance DB API for the same)
	AAR - I		P	P	P			

Binding Service Truth table when receiving AAR-U

Table 3-4 Binding Service Truth table when receiving AAR-U

S.No.	Message	IPv6/IPv4	APN/ DNN+SNSAAI	Session ID	Binding Lookup Query
1	AAR-U	P	P	P	Session-ID
2	AAR-U	P	P	P	IP Address + APN/ DNN+SNSSAI (If Rx Session Id not Found in Binding DB)
3	AAR-U	P		P	IP Address (If Rx Session Id not Found in Binding DB)

Note

- IP-CAN-Session Not available message is displayed when there is no suitable binding found in the binding service.
- UNABLE-TO-COMPLY message is displayed when request times out at Diameter Gateway.

Remote Cleanup of PDU Sessions when PDU Session Exceeds Limiting Number

In the event where the number of SM sessions per DNN exceeds the configured "Max sessions per DNN" value for a Binding service, Binding service triggers remote or local cleanup request towards SM service when the number of SM session per DNN exceeds the configured `Max sessions per DNN` value.

The deletion of stale binding session is controlled by a configurable binding flag. The binding flag "Max Session Cleanup Mode" can be set to either "Local" or "Remote" value.

- If the flag is set to "Local" PCF does not send terminate notification to SMF and deletes the session locally.
- If the flag is set to "Remote" PCF initiates terminate notification request toward SMF. PCF starts a timer to wait for SMF to send delete request to PCF based on `Force Delete On Expiry of Wait Timer` flag.
  - If delete request is received before timer expiry then deletes the session
  - If timer expires and delete request is not received PCF deletes the session

The timer value is configured in the PCF SM service Advanced setting page in CNC Console. Its default value is 30000 millisecond. This is considered only if "Remote" is selected in Binding service configuration.

On failure of PCF terminate notification request to SMF, PCF deletes the session based on the "Force delete on Error" flag.

3.5 Notifier Service

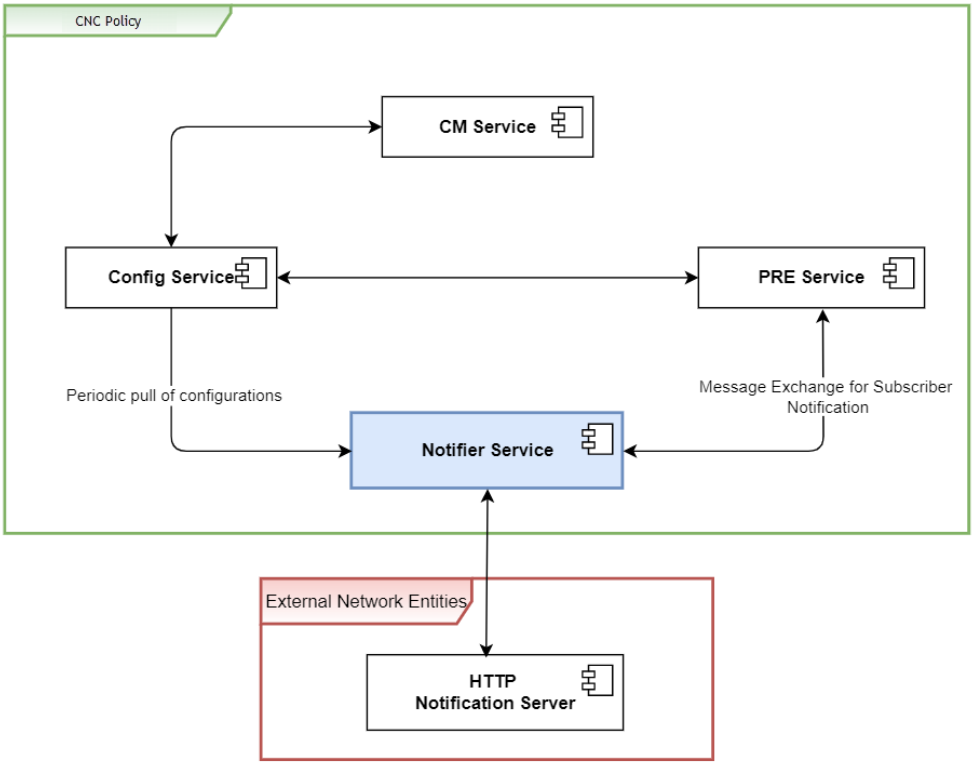
Policy Notifier service is an optional service that allows Policy to send notifications to the subscribers about their current data usage. This service is independent of deployment mode and can be enabled in all the supported deployment modes.

As part of subscriber quota management and based on the defined policies, the user is notified about their current usage at various levels. For example, when a user consumes 80% of the granted quota, a notification must be sent to notify the user about its usage. Similarly, when they reach or exceed the threshold, a notification must be sent informing the user about reduced quality of service or even suspension of service. With the Notifier service, Policy can send either HTTP notification or short messages (SMS) using SMPP to subscribers, based on the policies configured by operators, through an external notification server.

> **Note**
>
> In the current release, Policy supports subscriber notification only using HTTP and SMPP protocols.
> This feature is supported only for PCRF-Core call flows.

The following diagram shows how Notifier Service fetches configurations from Config service and on receiving notification request from PRE, sends out the notification to the subscriber via external notification server:

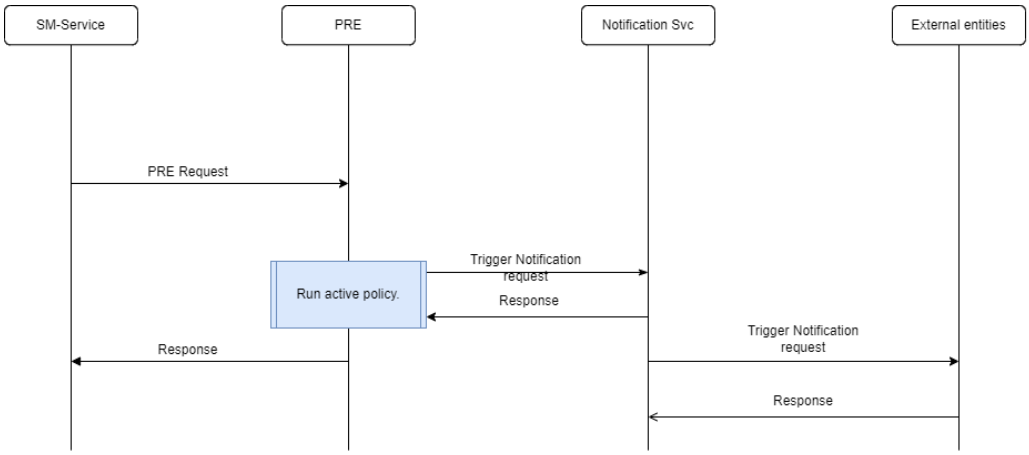
Figure 3-4 High Level Notifier Service Diagram



Call Flow

The following call flow diagram describes the notification call flow for SM service. As seen in the diagram, SM service sends a request to PRE that runs any given active policy and as a result triggers notification request to Notifier Service. Then, Notifier service sends a notification request towards the configured external server. On receiving success response, Notifier service sends the response to PRE, which in turn sends it to SM service.

Figure 3-5 Notifier Call Flow for SM Service



Enable

By default, the Notifier service is disabled. Thus, the user cannot access the configurations from GUI.

To enable the service, set the following parameter to true at the time of installing or upgrading Policy:

```
notifierServiceEnable
```

For information on how to set the parameter value, see *Oracle Communications Cloud Native Core Policy Installation, Upgrade and Fault Recovery Guide*.

Configure

Once the service is enabled, you should be able to see **Notifier** service menu under the **Service Configurations** on CNC Console for Policy.

To allow flexibility, operators can configure an HTTP request message including the destination, content, and other attributes. The static content of the notification is retrieved from Policy variables and Policy table.

For information on how to configure Notifier service, see Configuring Notifier Service.

Logs

Policy publishes logs for all Hypertext Transfer Protocol (HTTP) messages sent via Policy action.

3.6 NWDAF Agent

Policy integrates with Network Data Analytics Function (NWDAF) service to get analytics information. NWDAF can provide various analytics data, which PCF can consume and use to make policy decisions. For more information analytics data, see Oracle Communications Networks Data Analytics Function User Guide.

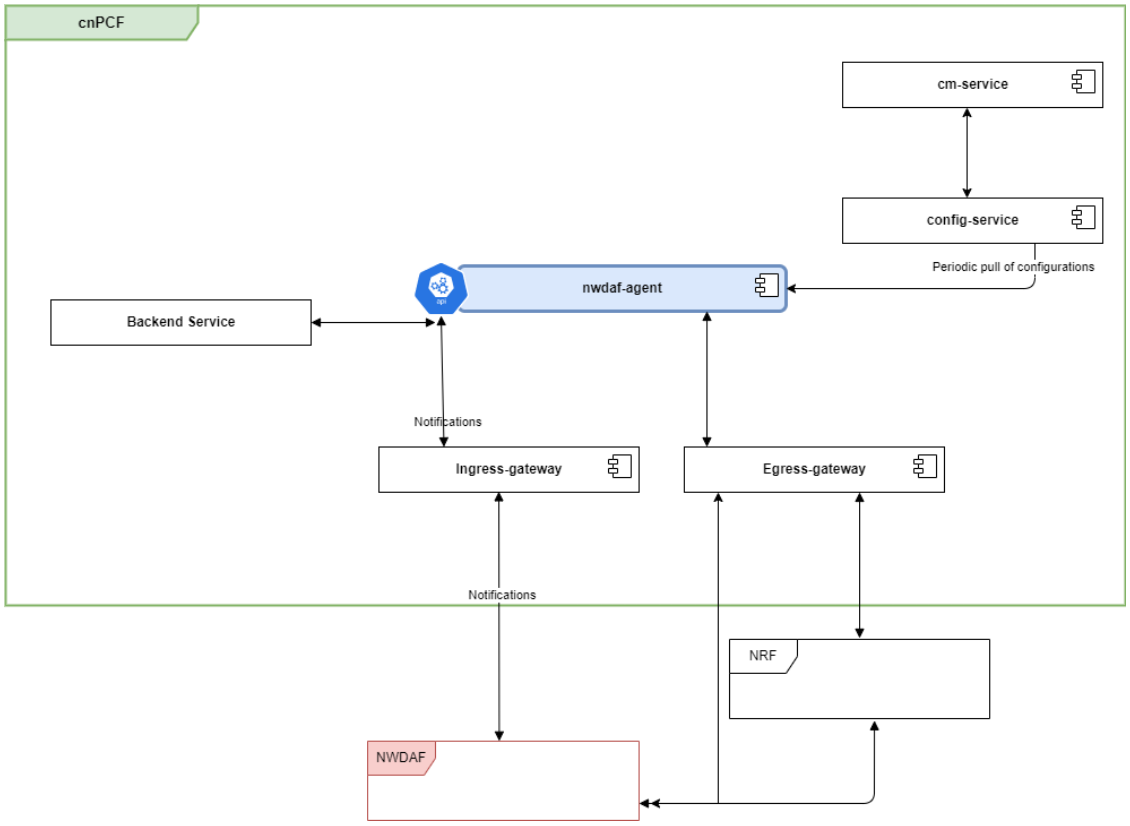
> **Note**
>
> Policy 22.4.0 includes phase one implementation of Policy integration with NWDAF for testing purposes.

It also queries Network Repository Function (NRF) client for fetching NWDAF information. For NWDAF, autonomous NRF discovery and static configuration is only supported, on-demand is not yet supported. For more information on autonomously discovered NF profiles for NWDAF, see Discovered NF Instances.

A 5G network contains a vast number of devices and sensors generating an enormous amount of data. The NWDAF Agent allows the Communications Service Providers (CSPs) to efficiently monitor, manage, automate, and optimize their network operations by the data collected and analytics generated across the network. It also helps the CSPs in achieving the operational efficiency and provides an enhanced service experience. Currently, NWDAF Agent supports slice load level analytics as part of the analytics data that is provided from NWDAF.

The following diagram describes the interaction with NWDAF Agent and Policy:

Figure 3-6 Interaction between Policy and NWDAF Agent



PCF supports the following 3GPP defined services for NWDAF Agent:

Table 3-5 3GPP defined services for NWDAF Agent

Service Operation Name	Description	Initiated By	Resource URI	HTTP Method or Custom Operation
GET	Get NWDAF Agent Service configuration	AM	{apiRoot}/oc-cnpolicy-configuration/v1/services/nwdafAgent	GET
PUT	Update NWDAF Agent Service configuration	AM	{apiRoot}/oc-cnpolicy-configuration/v1/services/nwdafAgent	PUT
GET	Export NWDAF Agent Service configuration	AM	{apiRoot}/oc-cnpolicy-configuration/v1/services/nwdafAgent/export	GET
POST	Import NWDAF Agent Service configuration	AM	{apiRoot}/oc-cnpolicy-configuration/v1/services/nwdafAgent/import	POST

For information on parameters of the 3GPP defined services for NWDAF Agent , see *Oracle Communications Cloud Native Core, Converged Policy REST API Specification Guide*.

Enable

By default, the NWDAF Agent is disabled. Thus, the user cannot access the configurations from GUI.

To enable the service, set the following parameter to true at the time of installing or upgrading Policy:

```
nwdafAgentServiceEnable
```

For information on how to set the parameter value, see *Oracle Communications Cloud Native Core Policy Installation, Upgrade and Fault Recovery Guide*.

Configure

Once the service is enabled, you should be able to see **NWDAF Agent** service menu under the **Service Configurations** on CNC Console for Policy.

For information on how to configure NWDAF Agent, see Configuring NWDAF Agent.

After the configuration you can use blockly to see the analytics data. For more information, see "Public Category" section in *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.

3.7 PCRF Core Service

The Policy solution supports the Gx reference point for provisioning and removal of PCC rules from the PCRF to the PCEF and the transmission of traffic plane events from PCEF to PCRF.

PCRF Core Service supports the following:

- IP-CAN session Establishment, Modification, and Termination Support
- Install, Modify, or Remove Predefined PCC rules
- Install, Modify, or Remove Dynamic PCC rules
- Gate function
- Charging related information support
- Integration with AF (over Rx)
- Presence Area Reporting (PRA) Support
- Time of the day procedures
- Sponsored data connectivity support
- NSA related enhancements for QoS (Quality of Service)
- UE IPv4, IPv6, and IPv4v6 support

For information about configuring PCRF Core service, see Settings.

Configurations to route messages to OCS or UDR through Diameter Gateway

> **Note**
>
> Configurations described in this section are required when PCRF Core uses UDR or OCS as a datasource.

When CNC Policy is deployed in cnPCRF or converged mode, PCRF Core service does not connect directly to OCS or UDR. Instead, PCRF Core establishes a connection with Diameter gateway - the front end for any Diameter traffic, and routing of messages destined for OCS/UDR has to go through the latter.

To allow PCRF Core to communicate with UDR and OCS through Diameter gateway, perform the following Diameter peer configurations:

1. Configure the OCS as OCS peer by putting the following values on the **Create Peer Node** page:

Table 3-6 Peer Node Configuration

Field Name	Description
Name	Name of the peer node. Example value: ocs
Type	Type of the peer node. Example value: ocs

Table 3-6 (Cont.) Peer Node Configuration

Field Name	Description
Reconnect Limit (sec)	The reconnect limit. This value must be configured as the Diameter peer configuration. Example value: `10`
Initiate Connection	Set to **true** to initiate the connection with peer node.
Port	OCS peer port detail. Example value: `8007`
Host	The OCS host IP/FQDN.
Realm	The realm detail of the OCS peer. Example value: `oracle.com`
Identity	The identity detail of the OCS peer. Example value: `ocs`

For more information, see Peer Nodes.

2. Configure the DataSource by providing the following values on Create Data Source page:

Table 3-7 Data Source Configuration

Field Name	Description
Name	Data source name. Example value: `ocs`
Description	Details about the data source.
Type	Data source type. Example value: `Sy`
admin state	Enable this switch.
Realm	The realm detail of the data source.
Role	Role of the data source. Example value: `Primary`
Timer Profile	Timer profile for the data source.
Primary Server	The primary data source server details. For Primary data source server, enter the following values: • **Identity**: Primary server identity. Example value: `oc-diam-gateway` • **Addr**: Load balancer IP of Diameter gateway. • **Port**: Port detail.

For more information, see Data Sources.

3. Configure the static routing table, using the configurations given in the following table:

Table 3-8 Diameter Route Table Configuration

Field Name	Description
Priority	Set the priority value. Example value: `1`

Table 3-8 (Cont.) Diameter Route Table Configuration

Field Name	Description
Name	Name of the route. Example value: `ocsroute`
Type	Route type Example value: `Realm`
Realms	The realm detail of the route. Example value: `oracle.com`
Application ID	Example value: `Sy`
Server Identifier	The value of this field is reference of Diameter peer configuration name. Example value: `ocs`

For more information, see Routing Table.

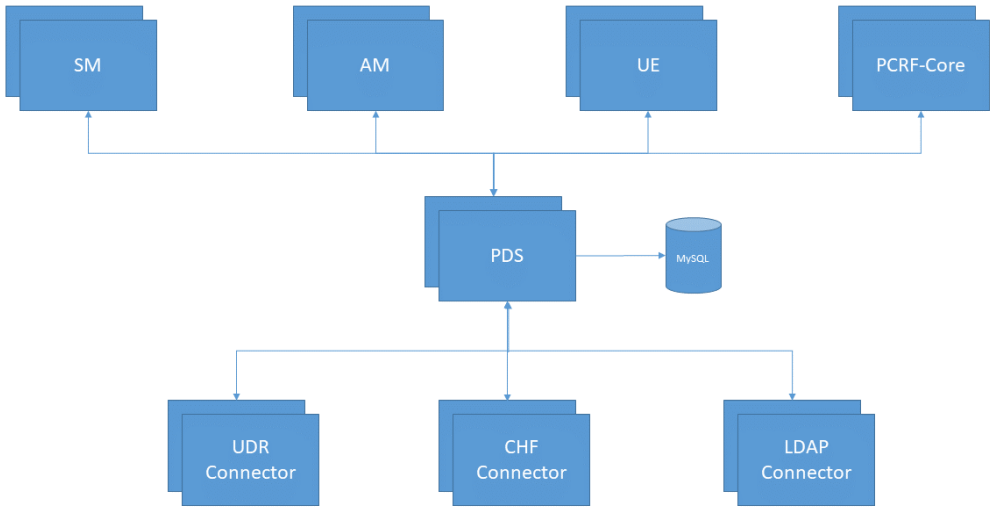
3.8 Policy Data Source (PDS) Service

Policy Data Source (PDS) Service

Policy Data Source Service (PDS) is an intermediate layer which interfaces Core Services (SM/AM/UE/PCRF-Core) with the protocol specific connector layers (UDR/CHF/LDAP Connectors). PDS also holds the subscriber information in its database when a new session gets established for a given subscriber and cleans-up the information when the last session termination happens. PDS is also responsible for storing the Subscriber State Variables except session level in its database and provides these variable information across multiple core services which requests the information.

You could define the workflows for PDS and the workflow would define the flow of request processing. This workflow is internal to the PDS project and should consult with engineering team for any update.

The following figure depicts the PDS architecture:



For configuring PDS service, see PDS Settings.

> **Note**
>
> Currently, PDS supports SUPI/GPSI as the search key. It does not support NAI as the search key.

UDR Connector- UDR Connector is a protocol specific layer which converts the request send by PDS to nUDR specific format and forwards to real UDR for subscription information. It also provides an ability to subscribe for profile change at UDR.

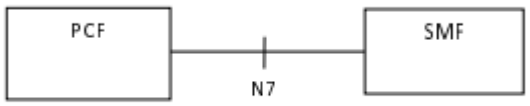
CHF Connector – CHF Connector is a protocol specific layer which converts the request send by PDS to nCHF specific format and forwards to actual CHF for fetching PolicyCounter Information. As per the standards, it automatically subscribes for profile change at CHF.
For configuring the UDR or CHF connector, see PCF User Connector.

3.9 Session Management Service

Oracle Communications Policy Control Function (PCF) implements policy control for session management for service data flows. PCF implements the N7 interface to trigger the Session Management (SM) policies towards Session Management Function (SMF). SMF controls the User Plane Function (UPF) . It translates policies received from PCF to a set of directives or information that can be understood by UPF and then forwards it to the UPF.

The following figure illustrates the communication between PCF and SMF over the N7 interface.

Figure 3-7 Communication between PCF and SMF over N7



Session Management Service supports the following:

- Enforcement control of policy decisions related to QoS, charging, gating, service flow detection, packet routing and forwarding, and traffic usage reporting.
- Enforcement of QoS, charging, gating, service flow detection, packet routing and forwarding, and traffic accounting and reporting policy decisions can be distributed among the UPF, Radio Access Network (RAN), and User Equipment (UE) depending on the policy type.
- Support for UE IPv4, IPv6, and IPv4v6

Oracle Communications PCF supports the following 3GPP defined services for Session Management:

Table 3-9 Session Management Services

Service Operation Name	Description	Initiated By	Resource URI	HTTP Method
Npcf_SMPolicyControl_Create	Request to create an SM Policy Association with the PCF to receive the policy for a PDU session	SMF	{apiRoot}/npcf-smpolicycontrol/v1/sm-policies	POST
Npcf_SMPolicyControl_Delete	Request to delete the SM Policy Association and the associated resources	SMF	{apiRoot}/npcf-smpolicycontrol/v1/sm-policies/{smPolicyId}/delete	POST
Npcf_SMPolicyControl_Update	Request to update the SM Policy association with the PCF to receive the updated policy when Policy Control Request Trigger condition is met	SMF	{apiRoot}/npcf-smpolicycontrol/v1/sm-policies/{smPolicyId}/update	POST
Npcf_SMPolicyControl_UpdateNotify	Update and/or delete the PCC rule(s) PDU session related policy context at the SMF and Policy Control Request Trigger information	PCF	{Notification URI}/update {Notification URI}/terminate	POST

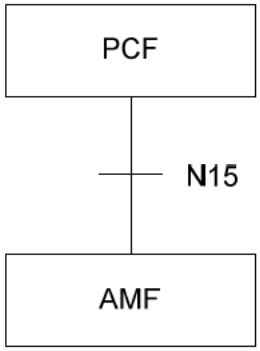
For configuring PCF Session Management service, see PCF Session Management.

3.10 UE Policy Service

Policy implements **PCF UE Policy** service to provision the UE policies, determined by PCF, to UE through AMF over the N15 interface.

The following figure describes the communication between PCF and AMF over the N15 interface:

Figure 3-8 PCF-AMF communication



PCF UE Policy service performs the following functions:

- Transfering UE Route Selection Policies (URSP) rules to UE
- Establishing the UE Policy Association requested by the NF service consumer
- Deleting the UE Policy Association requested by the NF service consumer
- Defining and delivering URSP message to UE via AMF using N1N2 message

Policy supports the following 3GPP defined service operations for the PCF UE Policy service:

Table 3-10 PCF UE Policy Services

Service Operation Name	Description	Initiated By	Resource URI	HTTP Method
Npcf_UEPolicyControl_Create	Creates a UE Policy Association	AMF	{apiRoot}/npcf-ue-policy-control/v1/policies/	POST
Npcf_UEPolicyControl_Update	Provides means for the NF consumer, that is, AMF to update an existing UE Policy association. AMF invokes this procedure only if PCF has subscribed to location change trigger.	AMF	{apiRoot}/npcf-ue-policy-control/v1/policies/{polAssoId}	POST
Npcf_UEPolicyControl_UpdateNotify	Notifies NF consumer about the update made to policy control request trigger(s) by PCF	PCF	{Notification URI}	POST

Table 3-10 (Cont.) PCF UE Policy Services

Service Operation Name	Description	Initiated By	Resource URI	HTTP Method
Npcf_UEPolicyControl_Delete	Provides means for the NF consumer to delete the UE Policy Association	AMF	{apiRoot}/npcf-ue-policy-control/v1/policies/{polAssoId}	DELETE
N1N2MessageSubscribe	Creates a subscription for N1 Message Transfer	AMF	{apiRoot}/namf-comm/<apiVersion>/ue-contexts/{ueContextId}/n1-n2-messages/subscriptions	POST
N1N2MessageUnSubscribe	Deletes an existing subscription for N1 Message Transfer	AMF	{apiRoot}/namf-comm/<apiVersion>/ue-contexts/{ueContextId}/n1-n2-messages/subscriptions/{subscriptionId}	DELETE
N1N2MessageTransfer	Transfers an N1 message (NAS message) to be delivered to the UE	PCF	{apiRoot}/namf-comm/<apiVersion>/ue-contexts/{ueContextId}/n1-n2-messages	POST
N1N2MessageNotify	Indicates status of an N1N2 Message Transfer	AMF	{apiRoot}/v1/ue-contexts/{ueContextId}/n1-n2-messages/notify	POST
N1N2MessageFailureNotify	Indicates that N1N2 message has failed to deliver to UE	AMF	{apiRoot}/v1/ue-contexts/{ueContextId}/n1-n2-messages/txfailure-notify	POST

For information on how to configure PCF UE Policy, see PCF UE Policy Service.

3.10.1 UE Policy Enhancements

UE Service in PCF is responsible for handling the User Equipment (UE) related procedures as described in 3GPP TS 29.525. This primarily includes delivery of UE Route Selection Policy (URSP) rules to the UE. URSP rules can either be locally configured at PCF or can be provided by the UE Route Selection Policy (URSP). The current implementation of PCF only supports local configuration of URSP rules, however UDR may send the UPSI codes to indicate to PCF which rules to deliver. PCF delivers URSP rules to UE via AMF on the N15 interface. PCF consumes the Namf-comm service provided (produced) by AMF on the N15 Namf interface.

AMF Selection for Namf-comm Subscription

Policy discovers the producer AMFs from NRF either using GUAMI or using the **AMF Set ID** and the **AMF Region ID**.

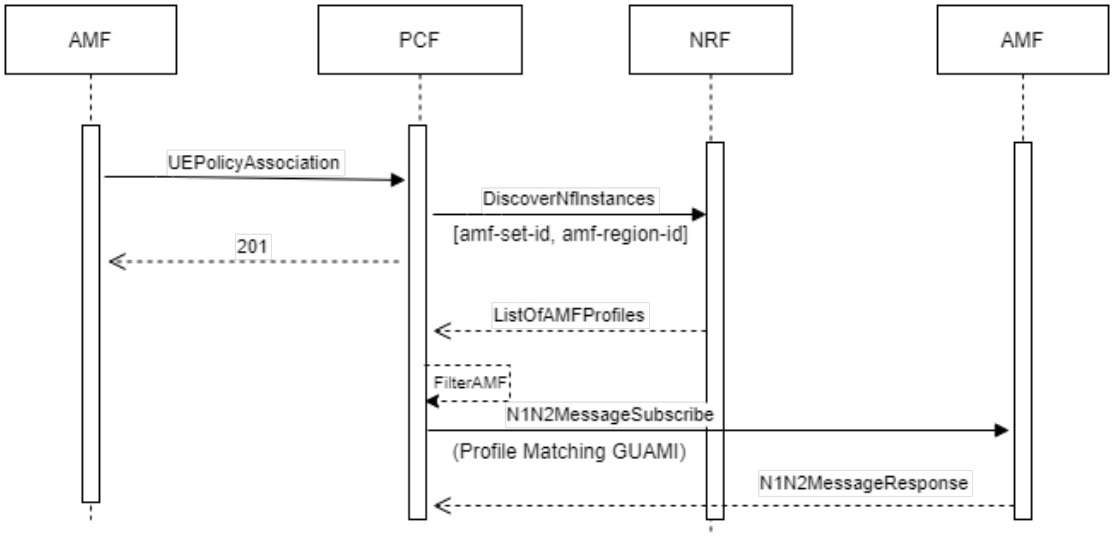
You can set the **AMF Discovery criteria** using CNC console.

1. Log in to CNC Console as an Administrator.
2. Navigate to **PCF UE Policy page** under **Policy**.
3. Under **AMF** section, set the value of **AMF Discovery criteria** to either **GUAMI** or **SetID and RegionID**.
   - By default, the value of **AMF Discovery criteria** parameter is set to **GUAMI**.

     With this default configuration, Policy sends GUAMI to NRF to receive the list of AMF profiles. From the list of AMF profiles received from NRF, Policy selects one of the AMFs as the producer AMF.
   - If the value of **AMF Discovery criteria** parameter is set to **SetID and RegionID**, Policy extracts the AMF SetID and AMF RegionID from GUAMI and uses these IDs to discover the producer AMF as explained below.

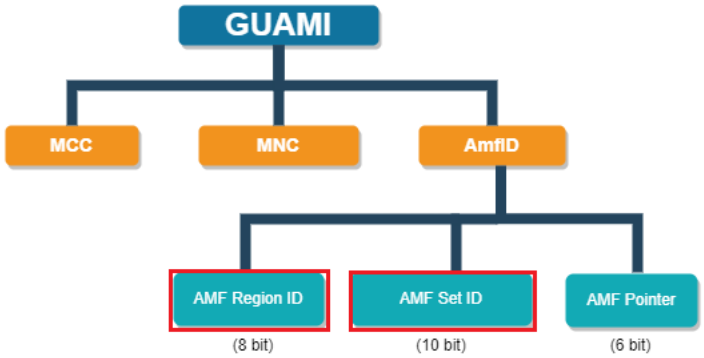
> **Note**
>
> GUAMI is the default configuration for all existing deployments. With this default configuration, the upgrade support for existing customers will also be acheived and any new UEPolicyAssociation Request will not impact the existing behavior of AMF discovery based on **GUAMI**.

Figure 3-9 Selecting Producer AMF using AMF SetID and AMF Region ID



1. Policy receives UEPolicyAssociationRequest from Consumer AMF.
2. If **AMF Discovery criteria** is set to **SetID and RegionID**, Policy parses the GUAMI to fetch the `AMF SetID` and `AMF RegionID`.

Figure 3-10 Parsing GUAMI



Sample GUAMI:

```
"guami": {
        "plmnId": {
            "mcc": "450",
            "mnc": "05"
        },
        "amfId": "010041"
    }
```

In the above example, Policy extracts the following details from GUAMI:

`amfid` = 010041

`AMF RegionID` (first 8 bits): 01

`AMF SetID` (next 10 bits): 001

3. Policy includes the extracted **AMF SetID** and **AMF RegionID** in a query parameter (`amf-set-id, amf-region-id`) and queries the NrfClientConnector through NRF discovery URI `/nnrf-disc/v1/nf-instances`.
4. Policy receives a list of AMF Profiles as a response from NRF.
5. Policy selects the target AMF from the list of AMF Profiles by matching GUAMI received in UEPolicyAssociationRequest with the GUAMI(s) present in the list of AMFProfiles returned from NRF.
   a. If the above match results in a single AMF instance, Policy uses that AMF for the Subscription Request required for UE Policy Transfer.
   b. If there is no match, Policy applies the filter for Priority/Capacity Load to all the GUAMI's received in AMFProfile to identify the relevant AMF.
   c. If there are more than one AMF profiles, Policy applies the filter for Priority/Capacity Load to select the relevant AMF.
6. Policy sends a N1N2MessageSubscribe to the selected producer AMF and receives a response.
7. Policy responds the consumer AMF with a 201 message.

UE Policy Delivery Rules

The following rules are defined in 3GPP TS 29.525 with respect to UE Policy (URSP) delivery:

- The PCF shall only send "MANAGE UE POLICY COMMAND" messages below a predefined size limit.
- The PCF may deliver the UE policy to the UE in several "MANAGE UE POLICY COMMAND" messages.

The UE policy is divided into policy sections for fragmented delivery and subsequent partial updates of UE policies. Such policy sections may be predefined in the PCF, may be retrieved by the PCF from the UDR as specified in 3GPP TS 29.519, or maybe dynamically generated by the PCF, but shall comply with the rules below. If the predefined size limit is observed, the PCF may combine several policy sections into one "MANAGE UE POLICY COMMAND" message.

The following rules apply for policy sections:

- The size must be below the predefined size limit.
- The policy section must contain complete URSP rule(s), and no fractions of such rules.
- The policy section may contain a small number of policies, for example, URSP rule(s), to ease a subsequent partial update of UE policies.
- A single PLMN should provide the entire content of a policy section.

Each UE policy section is identified by a UE policy section identifier (UPSI). The UPSI is composed of two parts:

1. The PLMN ID part containing the PLMN ID for the PLMN of the PCF which provides the UE policies.
2. An UE policy section code (UPSC) containing a unique value within the PLMN selected by the PCF.

Fragmented URSP Delivery

The PCF may deliver the UE policy to the UE in several "MANAGE UE POLICY COMMAND" messages.

After sending a "MANAGE UE POLICY COMMAND" message, the PCF shall wait for a related confirmation in a "MANAGE UE POLICY COMPLETE" message or failure indication in a "MANAGE UE POLICY COMMAND REJECT" message. When no such message is received until the expiry of a supervision timer specified in Annexure D of 3GPP TS 24.501, or when a failure indication is received, the PCF should resend related instructions for the policy sections.

When UE Policy is installed with fragmentation, policy table do not support URSPs installation. Policy supports this feature implementation, by providing URSP data type and URSP List data type as column in the policy table. On the Policy Console, this column data types is added in the Create Policy Table page. Two Blocklys namely, URSP List data type and URSP data type is used by the create Policy table blocks.

For detailed information about configuring the policy blocks for UE Policy service type, see *Oracle Communication Cloud Native Core, Converged Policy Design Guide*.

Manage UE Policy Command Reject

In case of PCF receiving "MANAGE UE POLICY COMMAND REJECT", policy tries to re-transmit the N1N2Message for all the rejected UPSIs only. PCF will decode UE Policy Manage Command Reject and will resend the UPSIs as part of UE Policy Manage Command which

were rejected by UE. This is done by setting the label `On UE Policy Command Reject` to **Re-Transmit Rejected UPSIs** value from the drop down list in the CNC Console.

> **Note**
>
> AMF notifying REJECT messages to PCF-UE with encoded rejected UPSIs should adhere to specification specified under Annexure D of 3GPP TS 24.501.

PCF handles the following N1N2Message transfer failures:

- On T 3501 Timer Expiry
- On Transaction Failure Notification
- On UE Policy Command Reject

Using CNC Console `N1 Message Retransmission Settings` group in `PCF UE Policy` page, the user can configure the UEPolicy `maximum number of retransmissions` and the actions to be taken to handle each failures separately.

For UE Policy configurations on CNC Console, see PCF UE Policy Service

Support of policy evaluation on AMF Notification on N1Notify and N1N2NotifyFailure

Currently, PCF does not evaluate Policies after N1 message notify or N1N2Transfer failure notification flow in PCF UE service. Thus PCF does not allow the user to take action dynamically and allows only static action such as retransmit, skip the fragment, or ignore.

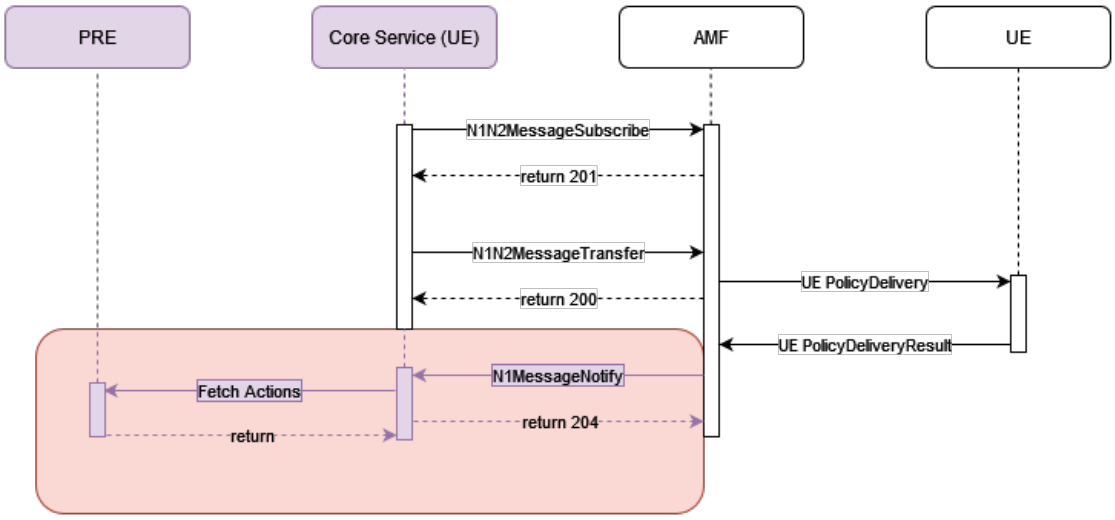
This feature enables the PCF user to write policies using Policy blockly. PCF evaluates the policies and takes actions after receiving N1Message Notify messages with MANAGE UE POLICY COMPLETE or MANAGE UE POLICY COMMAND REJECT messages from AMF , when UPSI's are installed. PCF UE service decodes the N1 Notify message that it receives from AMF during the rejection of URSP's by UE.

The subscription and transfer process are managed by the retry mechanism. In the notification flow, when the UE service receives a notification from AMF (N1MessageNotification), PRE is included both in the event of success or failure, allowing for subsequent installation attempts.

- In case of `MANAGE UE POLICY COMPLETE` message, PCF UE does not send the list of rejected UPSIs or URSPs since the transaction was completed successfully.
- In case of `MANAGE UE POLICY COMMAND REJECT` message, PCF UE sends the list of rejected UPSIs and URSPs to the PRE so that policy evaluation can be done using this information.
- In case of `N1N2MessageTransferFailure` message, PCF UE does not send the list of rejected UPSIs or URSPs as this is the a HTTP error with AMF and there is no information about rejected UPSIs or URSPS. PCF UE sends the cause of the failure to PRE.

> **Note**
>
> The blockly Policy created by considering the course of action that needs to be taken in the event of an error, supersedes any configurations set in `N1 Message Retransmission Settings` of the `PCF UE Policy` page in CNC Console.

Figure 3-11 UE and PRE interaction



The policy writer can check if the message is "MANAGE UE POLICY COMPLETE" and then take actions like:

- Write successfully the installed UPSI as a string value in VSA and update it to UDR.
- Define remote state variables, set values to variables and write in VSA and update it to UDR.
- Write successfully the installed UPSIs and UEPolicySection in UDR using the 3GPP Attributes-Non Fragmented or Fragmented (at message level).

The policy writer can check if the message is "MANAGE UE POLICY COMMAND REJECT" and then take actions like:

- – Write successfully the installed UPSI as a string value in VSA and update it to UDR.
  - – Write successfully the installed UPSIs and UEPolicySection in UDR using the 3GPP Attributes.
  - – Retransmit the failed UPSIs or new UPSIs, based on conditions like PLMN, SUPI, the failed UPSIs.
  - – Skip the retransmit of UPSIs, based on conditions like PLMN, SUPI, the failed UPSIs.
  - – Abort the transaction of transferring the UEPolicy in N1N2Message based on conditions like PLMN, SUPI, the failed UPSIs.
  - – Define remote variables other than UPSI, set values to variables and write in VSA and update it to UDR.

Deletion of URSP Rules

UE service allows deletion of URSP rule(s) in subsequent call flows that were previously sent over N1N2 NAS message delivery. The UPSI containers are updated and resent with the same UE Policy Section Code (UPSC) code and PLMN ID. An UPSI if being retransmitted because of URSP deletion will not further fragment the remaining URSPs within that UPSI as there is change in the size of the UPSI. The UPSI container honors the precedence of URSP rules while delivering in multiple fragments.

> **Note**
>
> The operator can perform Install and Remove URSP rules in the same policy execution cycle.

Managing UE Policy Enhancements

This section explains the procedure to enable and configure the feature.

Enable

This forms Policy applications core feature functionality. You do not need to enable or disable this feature.

Configure Using CNC Console

Perform the feature configurations in CNC Console as described in PCF UE Policy Service section.

Configure Using REST API

Perform the feature configurations as described in "UE Policy" section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Document*

Configure Using Blockly

PCF UE services blockly's are described "UE Policy Blocks" section in *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.

Observability

Metrics:

Policy provides UE Policy metrics as described in UE Service Metrics section.

Maintain
If you encounter alerts at system or application levels, see Alerts section for resolution steps.

In case the alerts still persist, perform the following:

- Collect the logs: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

3.11 Usage Monitoring Service

Policy Usage Monitoring is an internal service that interacts with Policy PRE service to get usage monitoring related Policy decisions and sends notifications to the subscribers about their current data usage. This service is independent of deployment mode and can be enabled in all the supported deployment modes.

With the introduction of Usage Monitoring service, Policy can control usage monitoring support for a PDU Session. Usage is defined as either volume or time of user plane traffic.

Policy receives usage monitoring related information per APN and UE from the UDR, i.e. the overall amount of allowed resources (based either on traffic volume and/or traffic time) that are to be monitored for the sessions of a user, together with the corresponding remaining allowed usage related information. In addition, usage monitoring related information for Monitoring

key(s) per APN and UE may also be received from the UDR, together with the corresponding remaining allowed usage related information. For the purpose of usage monitoring control Policy requests the Usage report trigger and provides the necessary usage threshold(s), either volume threshold, time threshold, or both volume threshold and time threshold, upon which the PGW reports to Policy. Policy may request a usage report from the PGW.

Once Policy receives a usage report from the PGW, it deducts the value of the usage report from the remaining allowed usage for that APN.

Enable

By default, the Usage Monitoring service is disabled.

To enable the service, set the following parameter to true at the time of installing or upgrading Policy:

```
usageMonEnable
```

For information on how to set the parameter value, see *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.

> **Note**
>
> To enable Usage Monitoring for PCRF, the deployment must have a 5G UDR as user data for Usage Monitoring is read from 5G UDR only.

Configure

Once the service is enabled, you can configure the Usage Monitoring service under the **Service Configurations** on CNC Console for Policy.

For information on how to configure Usage Monitoring service, see Configuring Usage Monitoring.

4 Policy Features

These section explains Oracle Communications Cloud Native Core, Converged Policy features.

> **Note**
>
> The performance and capacity of the Policy system may vary based on the Call model, Feature/Interface configuration, underlying CNE, and hardware environment, including but not limited to the complexity of deployed policies, policy table size, object expression and custom json usage in Policy design.

4.1 Message Feed for SBI Monitoring

In order to enable correlation of the internal and external (request/response) messages for all the transactions initiated by the producer and consumer NFs, Policy supports copying the messages at Ingress and Egress Gateways.

This feature allows NFs using Ingress and Egress Gateways to report every incoming and outgoing message to Oracle Communications Network Analytics Data Director (OCNADD) monitoring system.

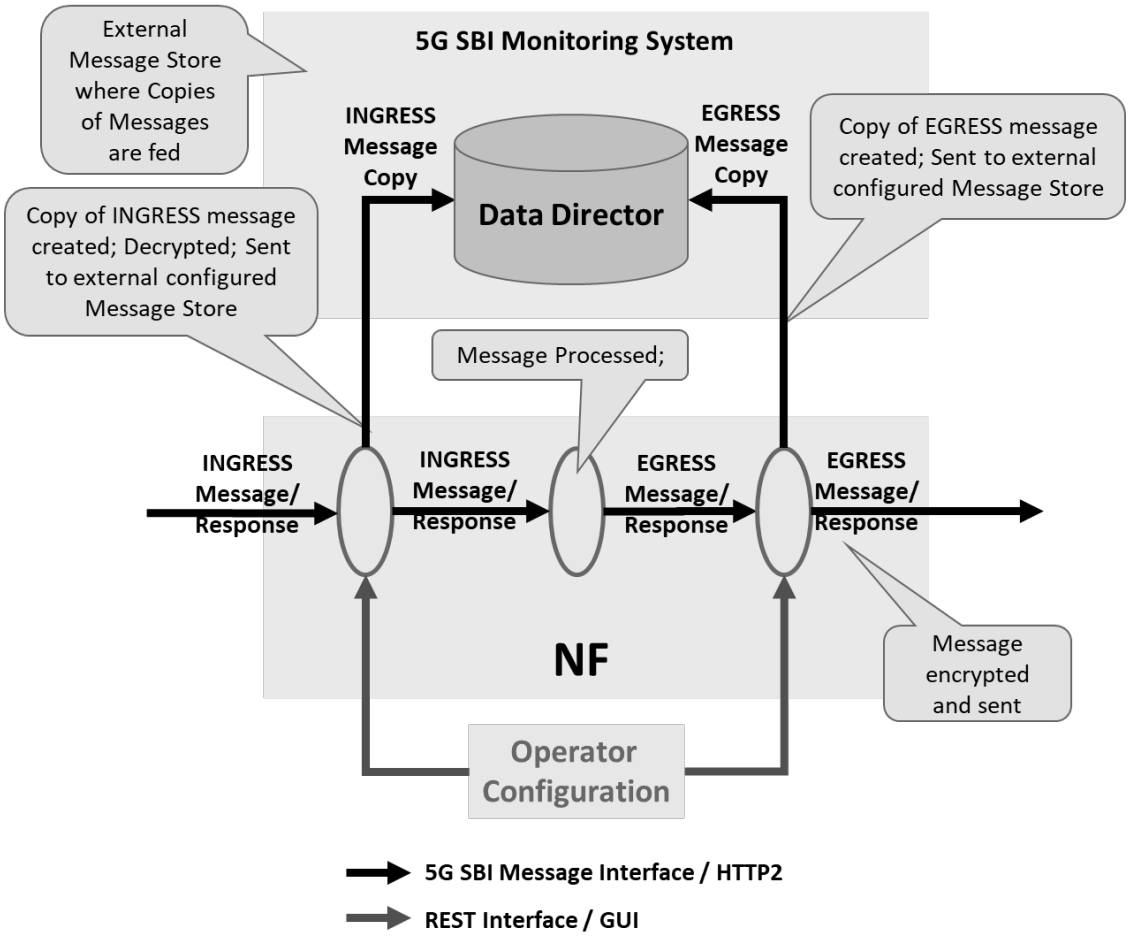
That is, OCNADD is a message store to keep a copy of each request and response processed through Ingress and Egress Gateways.

The insights on these messages enable NFs to integrate with external 5G SBI monitoring system for:

- Call Tracing / Tracking
- Live debugging

Architecture

Figure 4-1 Policy Message Feed Architeture



OCNADD is a Network Data Broker part of the Network Analytics suite of products. OCNADD receives network data traffic information from various sources such as 5G NFs and Non-5G Nodes and sends the data securely to subscribed consumer (3rd Party tools) after applying its powerful and configurable filtering, replication, and aggregation rule corresponding to subscribed consumers. For more information on OCNADD, see *Oracle Communications Network Analytics Data Director User Guide*.

5G NF Kafka Producer is used as the source to send the data stream towards OCNADD. The 5G NFs use integrated Kafka producer services to stream the 5G South Bound Interface (SBI) messages along with metadata added by NFs to OCNADD.

Managing Message Feed for SBI Monitoring

Enable

Message feed feature can be enabled using Helm parameters either at the time of Policy installation or during the software upgrade.

`ingress-gateway.message-copy.enabled` parameter is used to enable copying messages passing through Ingress Gateway.

`egress-gateway.message-copy.enabled` parameter is used to enable copying messages passing through Egress Gateway.

For more information, see *Configuring Ingress Gateway*, *Configuring Egress Gateway*, and *Configuring Kafka for NF Message Feed* sections in *Oracle Communications Cloud Native Core, Converged Policy, Upgrade, and Fault Recovery Guide*.

Configure

Message feed feature can be configured using Helm parameters either at the time of Policy installation or during the software upgrade.

For more information, see *Configuring Ingress Gateway*, *Configuring Egress Gateway*, and *Configuring Kafka for NF Message Feed* sections in *Oracle Communications Cloud Native Core, Converged Policy, Upgrade, and Fault Recovery Guide*.

SASL_SSL Configuration for Policy Message Copy

As there is no certificate-based client authentication required, a trustStore is created at Policy.

Policy contains placeholders to accept caroot certificates, which are then translated into trustStore using Gateway init-containers.

Policy uses native SSL functionality provided by Gateway services. SSL service block gets activated or used, when enableIncomingHttps is set to `true`. The same configuration is used for message copy SSL configuration too.

To configure only Policy-DD SSL communication without native SSL functionality, configure **caBundle** and **trustStorePassword** sections with appropriate secret configurations.

To use both native SSL functionality and Policy-DD SSL communication, add the caRoot certificate of Kafka broker to the existing caRoot certificate by appending Kafka broker ca certificate after the existing certificate.

1. Generate SSL certificates.

   > **Note**
   >
   > Creation process for private keys, certificates and passwords is based on discretion of user or operator.

2. Before copying the certificates to the secret, add the DD Root certificates contents into the CA certificate(**caroot.cer**) generated for NRF.

   > **Note**
   >
   > Make sure to add 8 hyphens "**-**" between 2 certificates.

   ```
   -----BEGIN CERTIFICATE-----
   <existing caroot-certificate content>
   -----END CERTIFICATE-----
   --------
   -----BEGIN CERTIFICATE-----
   <DD caroot-certificate content>
   -----END CERTIFICATE-----
   ```

3. Create a secret for authentication with DD.

To create a secret store the password in a text file and use the same file to create a new secret.

```
kubectl create secret generic ocingress-secret --from-
file=ssl_ecdsa_private_key.pem --from-file=rsa_private_key_pkcs1.pem --
from-file=ssl_truststore.txt --from-file=ssl_keystore.txt --from-
file=caroot.cer --from-file=ssl_rsa_certificate.crt --from-
file=ssl_ecdsa_certificate.crt --from-file=sasl.txt -n <namespace>

kubectl create secret generic ocegress-secret --from-
file=ssl_ecdsa_private_key.pem --from-file=ssl_rsa_private_key.pem --from-
file=ssl_truststore.txt --from-file=ssl_keystore.txt --from-
file=ssl_cabundle.crt --from-file=ssl_rsa_certificate.crt --from-
file=ssl_ecdsa_certificate.crt --from-file=sasl.txt -n <namespace>
```

4. Provide appropriate values for the SSL section.
SSL configuration:

```
service:
  ssl:

    privateKey:
      k8SecretName: ocegress-secret
      k8NameSpace: pcf
      rsa:
        fileName: rsa_private_key_pkcs1.pem
      ecdsa:
        fileName: ssl_ecdsa_private_key.pem

    certificate:
      k8SecretName: ocegress-secret
      k8NameSpace: pcf
      rsa:
        fileName: tmp.cer
      ecdsa:
        fileName: ssl_ecdsa_certificate.crt

    caBundle:
      k8SecretName: ocegress-secret
      k8NameSpace: pcf
      fileName: caroot.cer

    keyStorePassword:
      k8SecretName: ocegress-secret
      k8NameSpace: pcf
      fileName: key.txt

    trustStorePassword:
      k8SecretName: ocegress-secret
      k8NameSpace: pcf
      fileName: trust.txt

    initialAlgorithm: RS256
```

5. Configure the message copy feature.

```
messageCopy:
  enabled: true
  copyPayload: true
  topicName: PCF
  ackRequired: false
  retryOnFailure: 0
  security:
    enabled: true
    protocol: SASL_SSL
    tlsVersion: TLSv1.2
    saslConfiguration:
     userName: ocnadd
     password:
       k8SecretName: ocegress-secret
       k8NameSpace: pcf
       fileName: sasl.txt
```

6. Make sure to configure the correct SASL_SSL port in `kafka.bootstrapAddress` attribute. To get the correct value of this, refer to DD Kafka's Values.yaml file.

Observability

Metrics

The following metrics are used to count the ingress and egress messages at the gateways:

- oc_ingressgateway_msgcopy_requests_total
- oc_ingressgateway_msgcopy_responses_total
- oc_egressgateway_msgcopy_requests_total
- oc_egressgateway_msgcopy_responses_total

For more information, see:

- Ingress Gateway Metrics
- Egress Gateway Metrics

Alerts

The following alerts are raised when OCNADD is not reachable:

- INGRESS_GATEWAY_DD_UNREACHABLE_MAJOR
- EGRESS_GATEWAY_DD_UNREACHABLE_MAJOR

For more information, see

- INGRESS_GATEWAY_DD_UNREACHABLE_MAJOR
- EGRESS_GATEWAY_DD_UNREACHABLE_MAJOR

4.2 Traffic Segregation

This feature provides end-to-end traffic segregation to Policy based on traffic types. Within a Kubernetes cluster, traffic segregation can divide applications or workloads into distinct sections such as OAM, SBI, Kubernetes control traffic, etc. The Multus CNI container network

interface (CNI) plugin for Kubernetes enables attaching multiple network interfaces to pods to help segregate traffic from each Policy microservice.

This feature addresses the challenge of logically separating IP traffic of different profiles, which are typically handled through a single network (Kubernetes overlay). The new functionality ensures that critical networks are not cross-connected or sharing the same routes, thereby preventing network congestion.

With traffic segregation, operators can segregate traffic to external feeds and applications more effectively. Previously, all external traffic was routed through the same external network, but now, egress traffic from the Policy pods can be directed through non-default networks to third-party applications. This separation is achieved by leveraging cloud-native infrastructure and the load balancing algorithms in CNE.

The feature supports the configuration of separate networks, Network Attachment Definitions (NADs), and the Cloud Native Load Balancer (CNLB). These configurations are crucial for enabling cloud native load balancing, facilitating ingress-egress traffic separation, and optimizing load distribution within Policy.

Prerequisites

The CNLB feature is only available in Policy if CNE is installed with CNLB and Multus.

Cloud Native Load Balancer (CNLB)

CNE provides Cloud Native Load Balancer (CNLB) for managing the ingress and egress network as an alternate to the existing LBVM, lb-controller, and egress-controller solutions. You can enable or disable this feature only during a fresh CNE installation. When this feature is enabled, CNE automatically uses CNLB to control ingress traffic. To manage the egress traffic, you must preconfigure the egress network details in the `cnlb.ini` file before installing CNE.

> **Note**
>
> CNLB is supported only for IPv4 stack.

For more information about enabling and configuring CNLB, see *Oracle Communications Cloud Native Core, Cloud Native Environment User Guide*, and *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.

Network Attachment Definitions for CNLB

A Network Attachment Definition (NAD) is a resource used to set up a network attachment, in this case, a secondary network interface to a pod. Policy supports two types of CNLB NADs:

1. **Ingress Network Attachment Definitions**
   Ingress NADs are used to handle inbound traffic only. This traffic enters the CNLB application through an external interface service IP address and is routed internally using interfaces within CNLB networks.
   - **Naming Convention:**`nf-<service_network_name>-int`
2. **Egress Only Network Attachment Definitions**
   Egress Only NADs enable outbound traffic only. An NF pod can initiate traffic and route it through a CNLB application, translating the source IP address to an external egress IP address. An egress NAD contains network information to create interfaces for NF pods and routes to external subnets.

- **Requirements:** Destination (egress) subnet addresses are known beforehand and defined under the `cnlb.ini` file's `egress_dest` variable to generate NADs.
- **Naming Convention:**`nf-<service_network_name>-egr`

3. **Ingress/Egress Network Attachment Definitions**
   Ingress/Egress Network Attachment Definitions enable inbound/outbound traffic. An NF pod can initiate traffic and route it through a CNLB app, translating source IP address to an external egress IP address (defined under **cnlb.ini** file **egress_addr** variable). An Ingress/Egress Network Attachment Definition contains network information to create interfaces for NF pods and routes to external subnets. Even though an Ingress/Egress Network Attachment Definition enables outbound traffic, it also handles inbound traffic, so if inbound/outbound traffic is needed an Ingress/Egress Network Attachment Definition should be used.
   - **Requirements:** Source (ingress) and destination (egress) subnet addresses are known beforehand and defined under **cnlb.ini** file **egress_dest** variable to generate Network Attachment Definitions.
   - **Naming Convention:**`nf-<service_network_name>-ie`

Managing Ingress and Egress Traffic Segregation

Enable:

This feature is disabled by default. To enable this feature, you must configure the network attachment annotations in the custom values file.

Configuration

For more information about Traffic Segregation configuration, see " Configuring Traffic Segregation" section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide.*.

Observe

There are no Metrics, KPIs, or Alerts available for this feature.

Maintain

To resolve any alerts at the system or application level, see Alerts section. If the alerts persist, perform the following:

1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide*.
2. **Raise a service request**: See My Oracle Support for more information on how to raise a service request.

4.3 Support for Prevention of Requests Accumulation at Undertow Worker Queue

Prevention of Requests Accumulation at Undertow Worker Queue

The Congestion Control feature helps in pod protection during congestion state by rejecting the incoming requests based on their discard and request priorities. The request rejection are done by XNIO task threads. But during traffic bursts it is possible that XNIO task threads may get blocked. Due to this the rejection of requests because of pod congestion feature may not work

resulting in requests accumulation in undertow worker queue. Since the undertow worker queue is unbounded and the lot of requests have accumulated in its queue the pod may crash or restart due to out of memory issue.

The Undertow Queue Request Limiter functionality helps to handle this issue. The queue request limiter checks the undertow worker queue size, and if this size is greater than or equal to maximum acceptable size it starts rejecting the requests based on the configured discard priority and message priority. If the message priority is less than discard priority then that request is accepted,if not it will be rejected.

Currently this functionality is supported in SM, PDS, Binding, and Bulwark services. By default, this functionality is disabled for all of these services. The Users can enable this functionality, and also can customize the `maxAcceptRequestCount` size and `discardPriority` by adding the following parameters in custom `values.yaml` file for SM, PDS, Binding and Bulwark Services. Here is a sample Helm Configuration for PDS service:

```
policyds:
  serverHttpEnableBlockingReadTimeout: true
  undertow:
    queueRequestLimiter:
      enable: true
      discardPriority: 0
      maxAcceptRequestCount: 5000
```

> **Note**
>
> On enabling Undertow `queueRequestLimiter`, the requests are rejected without even passing to the Undertow worker queue and hence no application logic will be executed.

Managing Prevention of Requests Accumulation at Undertow Worker Queue

Enable

By default, this feature is disabled for the supported Policy services. You can enable this feature using helm parameters in custom `values.yaml` file.

For more information on Helm parameters, see "Configurations Parameters for Undertow Server Queue" section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*.

Observability

Metrics:

The following metrics are used to provide information about this feature:

- `occnp_undertow_queue_limiter_reject_request_total`
- `occnp_undertow_queue_limiter_accept_request_total`
- `occnp_undertow_queue_request_limiter_active_threads_count`

For information about the metrics, see Undertow Server Metrics.

4.4 Concurrency Handling at Bulwark Service to Reduce Processing Latency of Service

Policy supports the Bulwark service to handle the concurrent requests coming from other Policy services. Bulwark Service provides lock and unlock mechanism over a key (such as SUPI or GPSI) and allows only one notification at a time to proceed. For more information on handling concurrent requests, see Support for Concurrency Handling using Bulwark Service in Policy

Concurrency handling of different service requests for the same key at SM service or PDS (Policy microservices) has high latency in acquiring lock from Bulwark service, on reattempt, if lock was not acquired in the first attempt.

That is, when SM service or PDS sends a lock acquision request to Bulwark service and Bulwark service fails to acquire the lock within the `lockWaitTimeout` timer, SM service or PDS will have to wait until the `CONCURRENCY.LOCK_REQUEST_RETRY_BACKOFF` expires to resend a lock acquision request to Bulwark service again.

> **Note**
>
> `CONCURRENCY.LOCK_REQUEST_RETRY_BACKOFF` is an advanced settings key defined at SM service or PDS that indicates the amount of time after which the SM service or PDS can retry to gain the lock, incase the lock acquisition request fails.

SM service or PDS can fail to acquire the lock due to the following reasons:

- Fail to deliver lock request (for reasons due to transport issues, sidecar congestion, or bulwark pods restart).
- Lock is already acquired/busy.

SM service or PDS reattempts to acquire lock after `CONCURRENCY.LOCK_REQUEST_RETRY_BACKOFF` timer expires. SM service or PDS is not aware if the lock is available or if the lock that is released during this time and is available can be gained within the `CONCURRENCY.LOCK_REQUEST_RETRY_BACKOFF` timer expiry duration. This adds to the latency of processing concurrent requests.

> **Note**
>
> Currently, this feature is implemented only for SM service and PDS.

Policy reduces the latency in processing different concurrent service requests at SM service or PDS (Policy microservices) by acquiring the lock for the service request when available earlier than the `CONCURRENCY.LOCK_REQUEST_RETRY_BACKOFF` timer, instead of waiting for `CONCURRENCY.LOCK_REQUEST_RETRY_BACKOFF` timer to expire, if lock acquisition had failed previously.

Latency caused by `CONCURRENCY.LOCK_REQUEST_RETRY_BACKOFF` timer is addressed by using Long polling mechanism.

With this mechanism, SM service or PDS request lock from Bulwark service and wait for `lockWaitTimeout` or `RequestTimeout` period, for Bulwark service to respond. If Bulwark service is able to acquire a lock, it responds with acquired lock.

If Bulwark service cannot acquire lock, it uses the Long polling functionality to retry to acquire the lock for the given key (SUPI/GPSI). It starts polling to know when the exiting lock is released. For this purpose, Bulwark service creates a polling task and starts polling to know when the existing lock is released. Whenever an existing lock is released, it tries to acquire the lock for the same key.

If the lock acquisition is successful, it informs SM service or PDS with the acquired lock information, if `lockWaitTimeout` has not expired.

If the acquire lock request fails again, Bulwark service creates a new polling task and retries to acquire the lock within the `lockWaitTimeout` period. If `lockWaitTimeout` is expired, SM service or PDS will request after `CONCURRENCY.LOCK_REQUEST_RETRY_BACKOFF` timer again and the same steps to acquire a lock will be followed until maximum number of lock attempts is reached. Bulwark service then sends a failure notice to SM service or PDS .

The maximum number of lock requests to be allowed can be configured using **Maximum Pending Lock Requests allowed per Key** field under **Server Retry on-Already Locked Keys** section on **Bulwark Settings** page in CNC Console.

This feature can be enabled/disabled at SM service and/or PDS as well as at Bulwark service.

This feature can be enabled/disabled at SM service by configuring `CONCURRENCY.LOCK_RETRY_MODE` advance settings key on **Session Management service** page under **Service Configurations** section in CNC Console.

This feature can be enabled/disabled at PDS by configuring `CONCURRENCY.LOCK_RETRY_MODE` advance settings key on **PDS Settings** page under **Service Configurations** section in CNC Console.

This feature can be enabled and configured at Bulwark service using the fields under **Server Retry On Already Locked Keys** section on **Bulwark Settings** page in CNC Console.

If the feature is enabled at SM service and/or PDS, SM service and/or PDS should send the request initiation timestamp (oc-origination-request-timestamp) as header. Otherwise, the retry mechanism is not enabled for that request.

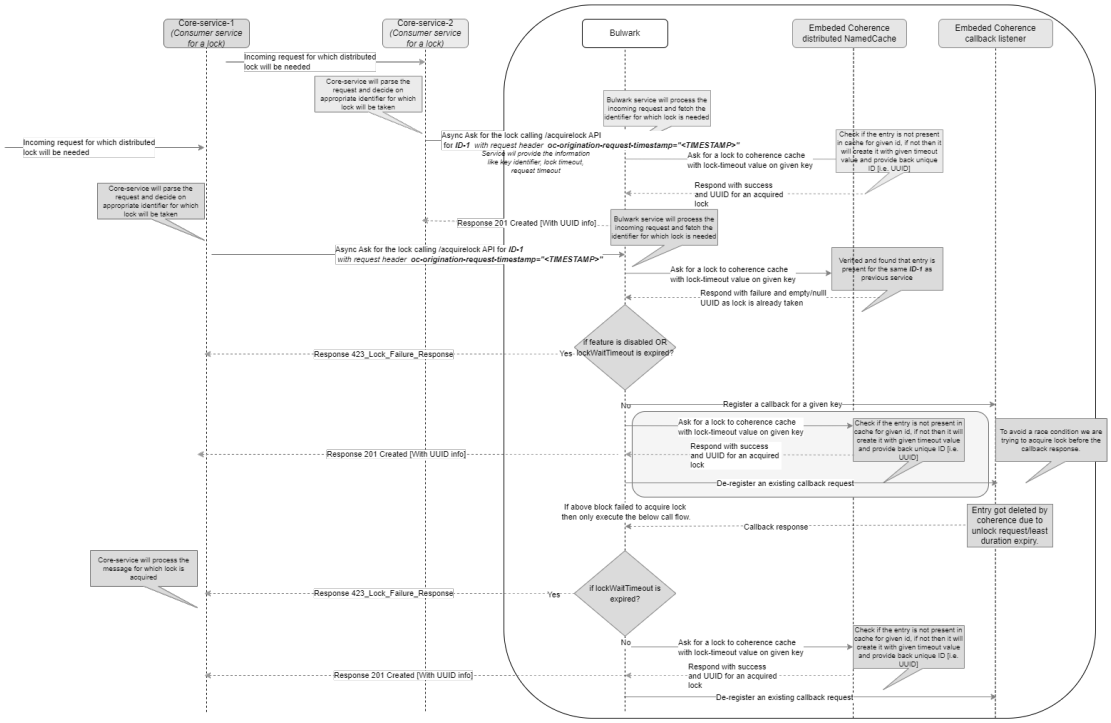
The time interval between which Bulwark service can poll to know if an existing lock is released can be configured using `polling.interval` Helm parameter.

The maximum number of times Bulwark service can retry to attempt the lock can be configured using `polling.maxLockAttempts` Helm parameter.

> **Note**
>
> Bulwark service supports Long polling functionality for both single and multi keys.

Call Flow

Figure 4-2 If lock is not available. Register for Long polling mechanism with the configured polling interval (20ms) and max lock re-attempts (5)



1. In case of multiple Create, Update, Delete, Notify, Update Notify requests for the same subscriber, SM service or PDS request a lock for the SUPI or GPSI value from the Bulwark service.
2. Bulwark service processes the lock request and fetches the identifier for which the lock is required.
3. Bulwark service tries to gain a lock from its cache. If there is no existing lock held by the given identifier, Bulwark service acquires a lock with the specified timeout value and a unique identifier (UUID) for the acquired lock.
4. Bulwark service responds to SM service or PDS (lock requesting service) with the lock details.
5. If Bulwark service fails to acquire the lock, it creates a new polling task and retries to acquire the lock within the `lockWaitTimeout` period.
6. If `lockWaitTimeout` has expired, Bulwark service sends a 423 lock failure response to SM service or PDS which is requesting the lock.
7. If `lockWaitTimeout` has not yet expired, Bulwark service creates a polling task with the default configuration (`polling.interval`=20ms and `polling.maxLockAttempts`=5).
8. After every 20ms, Bulwark service checks if an existing lock is released and reattempts to acquire the lock.

9. If the acquire lock request fails, Bulwark service continues to check every 20ms to know the availability of a lock and reattempts to acquire the lock until the maximum number of reattempts (`polling.maxLockAttempts`) reaches 5.
10. If Bulwark service acuires the lock during the reattempt, Bulwark service responds to SM service or PDS with the lock details.
11. If the maximum number of reattempts (`polling.maxLockAttempts`) reaches 5 and Bulwark service fails to acquire the lock, Bulwark service sends a 423 lock failure response to SM service or PDS which is requesting the lock.

> **Note**
>
> If feature is enabled at Bulwark service, if the `lockWaitTimeOut` is non-zero, and `oc-origination-request-timestamp` is present in the header, polling mechanism at Bulwark service will be enabled for that request.
>
> SM service or PDS must send the request initiation timestamp (`oc-origination-request-timestamp`) as header. Otherwise, retry mechanism can not be enabled for that request.
>
> Inoder to add the `oc-origination-request-timestamp` to the hearder in the lock request from SM service or PDS service to Bulwark service, the `LOCK_RETRY_MODE` must be either `SERVER_ONLY` or `CLIENT_AND_SERVER`.

Managing Notifications from Bulwark Service to the Service Requestor When Lock is Released

Enable

This feature can be enabled/disabled either at SM service and/or PDS as well as Bulwark service.

This feature can be enabled/disabled at SM service by configuring `CONCURRENCY.LOCK_RETRY_MODE` advance settings key on **Session Management service** page under **Service Configurations** section in CNC Console.

This feature can be enabled/disabled at PDS by configuring `CONCURRENCY.LOCK_RETRY_MODE` advance settings key on **PDS Settings** page under **Service Configurations** section in CNC Console.

This feature can be enabled and configured at Bulwark service using the fields under **Server Retry On Already Locked Keys** section in **Bulwark Settings** page in CNC Console.

For more information, see Bulwark Settings

Configure

This feature can be configured using Helm parameters, CNC Console, and REST API.

To configure the polling functionality by Bulwark service using Helm parameters, configure `polling.interval` and `polling.maxLockAttempts` parameters.

To configure using CNC Console:

- Configure the fields under **Server Retry on-Already Locked Keys** section on **Bulwark Settings** page in CNC Console.
  For more information, see Bulwark Settings.

- To indicate whether the lock retry must be triggered under which service, configure the **CONCURRENCY.LOCK_RETRY_MODE** advanced settings key available under **PCF Session Management** and **PDS** pages in CNC Console.
  For more information, see:
  - PCF Session Management
  - PDS Settings

To configure using REST API, configure the parameters in the following APIs:

- Acquire Lock
- Release Lock
- Server Retry on already locked keys

For more information, see *Oracle Communications Cloud Native Core, Converged Policy REST API Guide*.

Observability

Metrics

The following metrics are used for this feature:

- lock_request_total
- lock_response_total
- lock_collision_total

For more information, see Bulwark Metrics.

Alerts

The following alerts are used for this feature:

- LOCK_ACQUISITION_EXCEEDS_MINOR_THRESHOLD
- LOCK_ACQUISITION_EXCEEDS_MAJOR_THRESHOLD
- LOCK_ACQUISITION_EXCEEDS_CRITICAL_THRESHOLD

For more information, see List of Alerts.

4.5 Support for Optimizing Database Encoding in PCRF Core

This feature aims to optimize encoding and decoding the database fields of PCRF Cores services to reduce the size of data transferred during replication and improve the performance in the call flows.

PCRF Core stores and transfers the data in Java Script Object Notation (JSON) object. In JSON object the key/name describes the value, and the value is the actual data. At present, the PCRF Core JSON objects stored in the database is overly verbose, redundant keys and repetitive structures has increased the data size, causing longer data transfer times during replication.

The feature is implemented by:

- a data encoding technique that shortens all the overly verbose attribute names, and uses a mapping table that holds the long and short names.

- a data encoding technique that shortens all the overly verbose constants or default or enumeration attribute values, and uses a mapping table that holds the long and short values.
- to remove all unused and irrelevant JSON attributes from the value columns in gxsession, rxsession, sdsession tables.

Shorten JSON Attribute Keys

In the data encoding technique, every time before a record is stored in the gxsession, rxsession, sdsession tables it is encoded by mapping long attribute names to short names; when the same record is retrieved it is decoded by converting the short attribute names to its original long names.

For example: `"eventReportIndicationTriggerMask" shorten as "eRITM"`

Shorten JSON Attribute Values

In the data encoding technique, JSON attribute containing longer fixed or default or enumeration values are mapped to shorter values.

For example:

```
geographicLocationField": {"type":
"PCRFDiameterTrackingAreaEUTRANCellGlobalIdentifierImpl"
```

In the above example, the different values that "type" key can take and its mapping is shown below:

```
"PCRFDiameterTrackingAreaEUTRANCellGlobalIdentifierImpl",  we can map it to
"PDTAE";
"PCRFDiameterTrackingAreaIdentifierImpl", we can map it to "PDTAI";
"PCRFDiameterTrackingUpdateImpl", we can map it to "PDUI"
```

In PCRF Core services following Advanced keys are added to implement this feature:

- `DB.GX.DATA.ENCODING.Enabled`
- `DB.RX.DATA.ENCODING.Enabled`
- `DB.SD.DATA.ENCODING.Enabled`
- `DB.GX.ENCODING.MAP.Version`
- `DB.RX.ENCODING.MAP.Version`
- `DB.SD.ENCODING.MAP.Version`
- `DB.ENCODING.MAP.LIST`

Managing Support for Optimizing Database Encoding in PCRF Core

Enable

By default, this feature is disabled for PCRF Core service. You can enable this feature for gxsession, rxsession and rdsession using CNC Console.

Configure

To enable the feature using CNC Console, add the following Advanced Settings keys in the PCRF Core services and set their respective values to `true`:

- `DB.GX.DATA.ENCODING.Enabled`
- `DB.RX.DATA.ENCODING.Enabled`
- `DB.SD.DATA.ENCODING.Enabled`

For more information about the configurations, see PCRF Core Settings

Observability

Metrics

The following metrics are added to provide information about this feature:

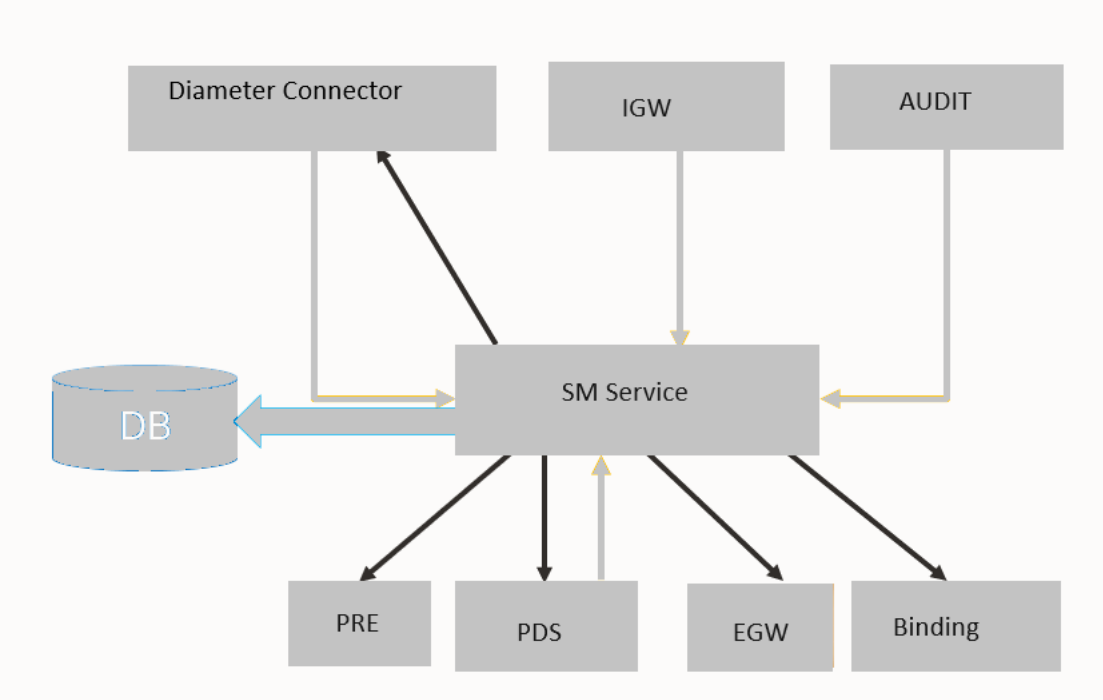
- `occnp_data_encoding_total`
- `occnp_data_decoding_total`
- `occnp_data_decoding_fail_total`
- `occnp_data_encoding_size_before_total`
- `occnp_data_encoding_size_after_total`

For information about the metrics, see PCRF Core Metrics section.

4.6 SM Service Pod Congestion Control

The Session Management (SM) service performs provisioning, update and removal of session related policies and PCC rules by PCF to Session Management Function (SMF). SM service interacts with other PCF services, as shown in the following diagram:

Figure 4-3 Policy Services and SM service interaction



- SMF interacts with SM service to Create/Update/Delete Policy associations.

- AF interacts with SM service to Create/Update/Delete App Sessions.
- PDS interacts with SM service to notify user data change and UDR subscription failures.
- Audit service interacts with SM service to process stale data.

At times, an excessive traffic from these services toward SM service can be observed in the network, which can result in a high CPU utilization, high memory utilization. This can cause performance degradation in SM service responses and eventually reach a state of service unavailability. Congestion control helps to identify such conditions and invoke load shedding rules to address these situations when these load conditions persist.

The SM service pod congestion control mechanism ensures consitent service availability to its consumer.

The pod congestion control mechanism involves:

1. Determining Pod Congestion State
2. Triggering Pod Congestion Control

Determining Pod Congestion State

In the pod congestion control mechanism, each SM service pod monitors its congestion state. The congestion control works at 5 levels or states:

- NORMAL
- DANGER_OF_CONGESTION (DOC)
- CONGESTION_L1
- CONGESTION_L2
- CONGESTED

The pod's congestion state is decided based on CPU consumption and Queue.

1. **CPU**: The CPU usage for congestion state is calculated by comparing the CPU usage of the container (monitored using cgroup parameter, `cpuacct.usage`, which provides current cpu usage in nanoseconds) with the configured threshold.
2. **Queue**: For the DOC, CONGESTION_L1, CONGESTION_L2, and CONGESTED pod states, compare the number of pending messages in the queue with the configured pending messages threshold.

The SM service pod's congestion states and their default congestion parameters, CPU, and Queue counts are provided in the following table:

Table 4-1 SM Service Congestion States

Congestion States	CPU (%)	Queue Count (Pending Requests)
DANGER_OF_CONGESTION (DOC)	77	140
CONGESTION_L1	78	160
CONGESTION_L2	79	180
CONGESTED	80	200

To avoid toggling between these states due to traffic pattern, it is required for the pod to be in a particular state for a given period before transitioning to another state. The below configurations are used to define the period that the pod has to be in a particular state for:

- `stateChangeSampleCount`: This REST API parameter can be configured to specify after how many continuous intervals, the pod state can be changed. This value can range from 1 to 2147483647.
- `stateCalculationInterval`: This REST API parameter can be configured to specify the time duration or interval, after which the pod congestion state will be re-verified. This interval is configured in milliseconds and can range from 50 to 2147483647.

Triggering Pod Congestion Control

Every time SM service receives requests from other services, it checks for the current congestion state of the pod. The Congestion Control mechanism is triggered if the pod's congestion state is in DOC or Congested_L1 or Congested_L2 or Congested.

The requests to the SM service might have priority included as `oc-message-priority` attribute in the request header. The priority value ranges between 0 to 100 with 0 being the highest and 100 being the lowest priority.

> **Note**
>
> Currently, the downstream services do not propagate the `oc-message-priority` header to SM service and will be implemented in future releases.

Priority-Based Load Shedding

Based on the pod's current congestion state, a load shedding rule is applied to perform priority-based load shedding. The load shedding rule is based on message priority. For example, when the SM service pod state is CONGESTED and thepriority of discard messages is 30, then it determines if the message with the assigned priority should be rejected or accepted.

These rules get configured per congestion state. If there are no rules configured for a congestion state, then SM service accepts the request as a default behavior. The user can configure the result codes for the rejected requests when configuring the load rules. The default result code is `503 Service Unavailable`.

The default load shedding rules for SM service:

```
- state: DANGER_OF_CONGESTION
      discardPriority: 27
- state: CONGESTION_L1
      discardPriority: 19
- state: CONGESTION_L2
      discardPriority: 17
- state: CONGESTED
      discardPriority: 15
```

When SM service is in congestion state, its response can be configured using SM service advanced settings in CNC Console, using the key CONGESTION_RESPONSE_CODE. This key is used to configure the response code of the messages that are rejected by the SM service due to pod's congestion state. By default, SM Service responds with a response code of 503. The response code configured should be 5xx error status only. Following is the list of configurable keys that can be added to set the message priority:

Table 4-2 Configuring Message Priority

Key	Default Value	Allowed Values
SM.UPDATE.EVENT.SUBS.PRIORITY	18	0-31
SM.CREATE.PRIORITY	24	0-31
SM.SUB.FAIL.NOTIFY.PRIORITY	26	0-31
SM.USER.SERVICE.NOTIFY.PRIORITY	18	0-31
SM.UPDATE_PRIORITY	18	0-31
SM.REAUTH_PRIORITY	20	0-31
SM.DELETE.PRIORITY	16	0-31
SM.POLICY.CLEANUP.PRIORITY	20	0-31
SM.APP.SESSION.CREATE.PRIORITY	24	0-31
SM.APP.SESSION.DELETE.PRIORITY	16	0-31
SM.APP.SESSION.CLEANUP.PRIORITY	20	0-31
SM.AUDIT.NOTIFY.PRIORITY	30	0-31
SM.GET.APP.SESSION.PRIORITY	28	0-31
SM.GET.ASSOC.PRIORITY	28	0-31
SM.GET.SUBSCRIBER.SESSIONS.PRIORITY	28	0-31
SM.GET.ASSOC.QUERY.PRIORITY	28	0-31
CONGESTION_RESPONSE_CODE.	503	5xx

Managing SM service Pod Congestion Control

Enable

By default, the Pod Congestion Control is disabled for SM service. You can enable this feature using CNC Console or REST API for Policy.

Configure Using CNC Console

To enable the feature using CNC Console, set the **Enable** parameter in **Settings** page under **Congestion Control** for **Overload and Congestion Control Configurations**.

To configure SM service pod congestion control feature in CNC Console, see Congestion Control section.

To configure SM service pod congestion control discard messages in CNC Console, see the Advanced Settings section in PCF Session Management section.

Configure Using REST API

Perform the feature configurations as described in "Congestion Control" section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Observability

Metrics:

The following metrics are used to provide information about this feature:

- `occnp_pod_congestion_state`
- `occnp_pod_resource_stress`
- `occnp_pod_resource_congestion_state`
- `pod_cong_state_report_total`
- `pod_resource_congestion_state_report_total`
- `http_congestion_message_reject_total`

For information about the metrics, see Pod Congestion Metrics.

Alerts

The following alerts are generated for this feature:

- POD_CONGESTED
- POD_CONGESTION_L2
- POD_CONGESTION_L1
- POD_DANGER_OF_CONGESTION
- POD_PENDING_REQUEST_CONGESTED
- POD_PENDING_REQUEST_CONGESTION_L2
- POD_PENDING_REQUEST_CONGESTION_L1
- POD_PENDING_REQUEST_DANGER_OF_CONGESTION
- POD_CPU_CONGESTED
- POD_CPU_CONGESTION_L2
- POD_CPU_CONGESTION_L1
- POD_CPU_DANGER_OF_CONGESTION

For more information about alerts, see Common Alerts.

Maintain

Warning logs are generated to indicate the congestion level. However, error logs are not generated when messages are rejected to avoid additional resource usage to write error logs.

If you encounter alerts at system or application levels, see Alerts section for resolution steps.

In case the alerts still persist, perform the following:

- Collect the logs: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide*.
- Raise a service request: See My Oracle Support.

4.7 Support for policyDecFailureReports Attribute

Policy Decision Error Handling is one of the 3GPP-defined features responsible for enabing the `policyDecFailureReports` attribute.

If the Policy Decision Error Handling is supported and the SMF receives one or more policy decisions and/or condition data which are not referred by any PCC rules or session rules as defined in 3GPP but the storage of the policy decisions and/or condition data was unsuccessful , the SMF behaves as follows:

- When the PCF receives update notify response or SM Update request, it considers all the instances of the policy decisions and/or condition data which are provisioned in the request message as removed from the SMF.
- When the PCF receives the response with HTTP "400 Bad Request" status code but the "policyDecFailureReports" attribute is not included, the PCF considers the request message as removed from the SMF.

The PCF does not remove the PCC rule in case the updatenotify request for rule remove failure. With introduction of the new flag **RULE.ENABLE_PCC_RULE_REMOVE_ON_FAILURE** set to true and if rule remove updatenotify request fails with "400 BAD Request" HTTP error, the association will be updated with PCC rule removed.

> **Note**
>
> If SMF responds with 200 error code and policyDecFailureReport includes the policyDecisionFailureCode for which the policy decision object were not sent will also be processed by PCF and the available policy decision object will be removed. Moreover, if SMF responds with 400 error code and policyDecFailureReport includes the policyDecisionFailureCode for which the policy decision object were not sent, then policy decision objects which were part of update notify request will be removed.

Managing Support for policyDecFailureReports Attribute

Enable and Configure

By default, this feature is not configured on the CNC Policy deployment. You can opt to configure the `policyDecFailureReports` attribute using the following ways in CNC Console.

- From the navigation menu under **Policy**, navigate to **Service Configurations** to select `PolicyDecisionErrorHandling` from the drop-down menu of **Override Supported Features** parameter.
- From the navigation menu under **Policy**, navigate to **Service Configurations** to add `RULE.ENABLE_PCC_RULE_REMOVE_ON_FAILURE` key in the **Advanced Settings**.

For more information about these configurations, see PCF Session Management.

Observe

The *occnp_feature_info_received_total* metric is is used to support `policyDecFailureReports` attribute. For more information, see SM Service Metrics.

4.8 Enhancements to Error Response

Policy sends error responses to consumer NFs due to some exceptions, such as signaling, validations, and internal errors. These error responses have payloads containing the problem title, status, details, and cause of the error that are used to investigate the error. The details section is now enhanced with application error IDs.

The error handling module gives provision to configure the error response dynamically and the same is responded when Policy is producer of the call flow.

With the enhanced error response mechanism, Policy sends additional information such as server FQDN, micro-service ID, error category, and application error ID in the `detail` attribute of the `ProblemDetails`. This enhancement provides more information about the error and troubleshoot them.

> **Note**
>
> As per the definition of `ProblemDetails` data type in 3GPP, the title and cause fields are optional and can be null or empty.

Application error ID follows the below format.

[EC] [NF ID] [Microservice ID] [Category] [Error ID]

An error code dictionary will be provided to identify the cause and possible solution of the error. For more details of the error code dictionaries for AM, UE, SM, UDR, CHF, and Binding services, see Error Code Dictionary.

Managing Enhancements to Error Response

This section explains the procedure to enable and configure the feature.

Enable

By default, this feature is disabled. The operator can enable this feature through the CNC Console configurations.

Configure

You can configure error handling functionality under **Error Handling** on CNC Console for Policy. For information about how to configure for AM, UE, SM, UDR, CHF, and Binding services in CNC Console, see Error Handling.

Observe

The following metrics have been added in AM, UE, SM, Binding, and User service (UDR and CHF) for this feature:

- error_handler_exec_total
- error_handler_in_total
- error_handler_out_total

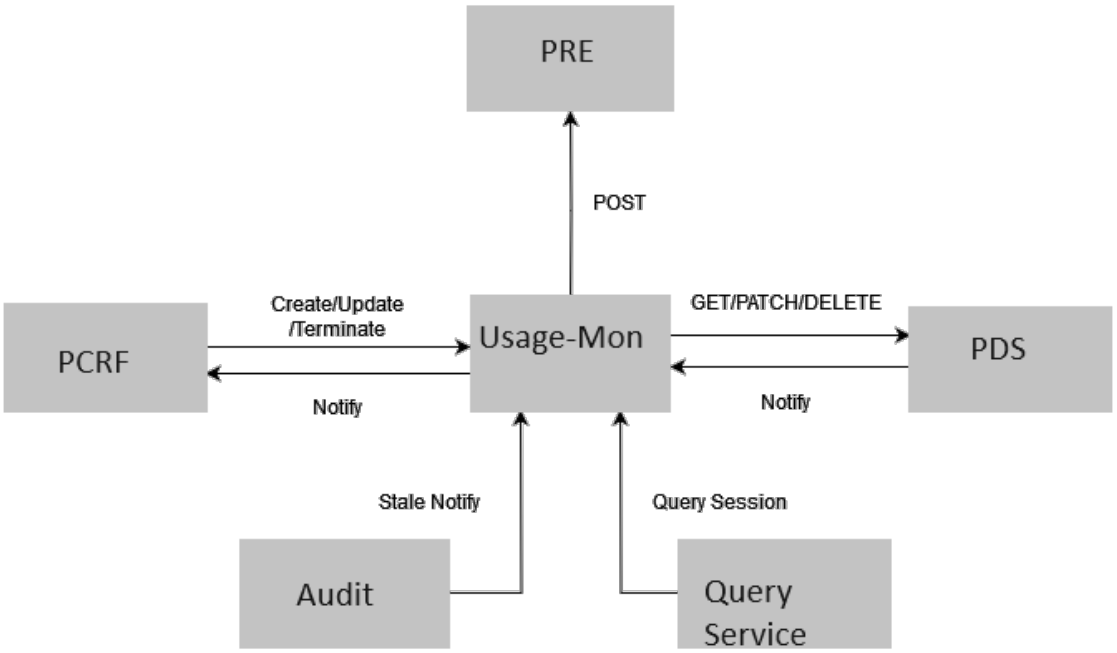
For more information, see CNC Policy Metrics.

4.9 Usage Monitoring Pod Congestion Control

The Policy Usage Monitoring service interacts with PRE and UDR services to get usage monitoring related Policy decisions and it also sends notifications to the subscribers about their current data usage.

Usage Monitoring interacts with other Policy services, as shown in the following diagram:

Figure 4-4 Usage Monitoring Service and other Policy services



- The PCRF core services send requests to Usage Monitoring service.
- Audit service sends requests to Usage Monitoring service for stale session notification.
- Usage Monitoring service sends requests to PRE for retrieving Policy decisions.
- Usage Monitoring service sends requests to PDS for Subscriber information.

At times, an excessive traffic from these services toward Usage Monitoring service can be observed in the network, which can result in a high CPU utilization, high memory utilization. This can cause performance degradation in Usage Monitoring service responses and eventually reach a state of service unavailability. Congestion control helps to identify such conditions and invoke load shedding rules to address these situations when these load conditions persist. The Usage Monitoring serivce pod congestion state is identified by CPU utilization and pending requests in the Queue.

The Usage Monitoring pod congestion control mechanism ensures consitent service availability to its consumer.

The pod congestion control mechanism involves:

1. Determining Pod Congestion State
2. Triggering Pod Congestion Control

Determining Pod Congestion State

In the pod congestion control mechanism, each Usage Monitoring service pod monitors its congestion state. The congestion control works at 5 levels or states:

- NORMAL
- DANGER_OF_CONGESTION (DOC)
- CONGESTION_L1
- CONGESTION_L2
- CONGESTED

The pod's congestion state is decided based on CPU consumption and Queue.

1. **CPU**: The CPU usage for congestion state is calculated by comparing the CPU usage of the container (monitored using cgroup parameter, `cpuacct.usage`, which provides current cpu usage in nanoseconds) with the configured threshold.
2. **Queue**: For the DOC, CONGESTION_L1, CONGESTION_L2, and CONGESTED pod states, compare the number of pending messages in the queue with the configured pending messages threshold.

The Usage Monitoring service pod's congestion states and their default congestion parameters, CPU, and Queue values are provided in the following table:

Table 4-3 Usage Monitoring Service Congestion States

Congestion States	CPU (%)	Queue (Pending Requests)
DANGER_OF_CONGESTION (DOC)		
CONGESTION_L1		
CONGESTION_L2		
CONGESTED		

To avoid toggling between these states due to traffic pattern, it is required for the pod to be in a particular state for a given period before transitioning to another state. The below configurations are used to define the period that the pod has to be in a particular state for:

- `stateChangeSampleCount`: This REST API parameter can be configured to specify after how many continuous intervals, the pod state can be changed. This value can range from 1 to 2147483647.
- `stateCalculationInterval`: This REST API parameter can be configured to specify the time duration or interval, after which the pod congestion state will be re-verified. This interval is configured in milliseconds and can range from 50 to 2147483647.

Triggering Pod Congestion Control

Every time Usage Monitoring service receives requests from other services, it checks for the current congestion state of the pod. The Congestion Control mechanism is triggered if the pod's congestion state is in DOC or Congested_L1 or Congested_L2 or Congested.

The requests to the Usage Monitoring service might have priority included as `oc-message-priority` attribute in the request header. If the prioirity is not included in the request, then the default priorities defined in the Usage Monitoring, CNC Console is considered. The priority value ranges between 0 to 31 with 0 being the highest and 31 being the lowest priority.

> **Note**
>
> Currently, the downstream services do not propagate the `oc-message-priority` header to Usage Monitoring service and will be implemented in future releases.

Priority-Based Load Shedding

Based on the pod's current congestion state, a load shedding rule is applied to perform priority-based load shedding. The load shedding rule is based on message priority. For example, when the Usage Monitoring service pod state is CONGESTED and the priority of discard messages is 30, then it determines if the message with the assigned priority should be rejected or accepted.

These rules get configured per congestion state. If there are no rules configured for a congestion state, then Usage Monitoring service accepts the request as a default behavior. The user can configure the result codes for the rejected requests when configuring the load rules. The default result code is `503 Service Unavailable`.

The default load shedding rules for Usage Monitoring service:

```
- state: DANGER_OF_CONGESTION
      discardPriority:
- state: CONGESTION_L1
      discardPriority:
- state: CONGESTION_L2
      discardPriority:
- state: CONGESTED
      discardPriority:
```

When Usage Monitoring service is in congestion state, its response can be configured using Usage Monitoring service **Message Default Priority for Congestion Control** settings in CNC Console, using the key CONGESTION ERROR CODE. This key is used to configure the response code of the messages that are rejected by the Usage Monitoring service due to pod's congestion state. By default, Usage Monitoring service responds with a response code of 503. The response code configured should be 5xx error status only. Following is the list of configurable Usage Monitoring API message default priorities for Congestion Control:

Table 4-4 Configuring Message Priority

Key	Default Value	Allowed Values
UM Session Create API	20	0-31
UM Session Update API	17	0-31
UM Session Terminate API	15	0-31
UM Session Notify API	17	0-31
UM Session Audit Subscriber API	31	0-31
UM Session Search Subscriber API	30	0-31
UM Session Audit Notify API	31	0-31
Congestion Error Code	503	5xx

Managing Usage Monitoring service Pod Congestion Control

Enable

By default, the Pod Congestion Control is disabled for Usage Monitoring service. You can enable this feature using CNC Console or REST API for Policy.

Configure Using CNC Console

To enable the feature using CNC Console, set the **Enable** parameter in **Settings** page under **Congestion Control** for **Overload and Congestion Control Configurations**.

To configure SM service pod congestion control feature in CNC Console, see Congestion Control section.

To configure SM service pod congestion control discard messages in CNC Console, see Message Default Priority for Congestion Control fields in Configuring Usage Monitoring section.

Configure Using REST API

Perform the feature configurations as described in "Congestion Control" section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Observability

Metrics:

The following metrics are used to provide information about this feature:

- `occnp_pod_congestion_state`
- `occnp_pod_resource_stress`
- `occnp_pod_resource_congestion_state`
- `pod_cong_state_report_total`
- `pod_resource_congestion_state_report_total`
- `um_http_congestion_message_reject_total`

For information about the metrics, see Pod Congestion Metrics.

Alerts

The following alerts are generated for this feature:

- POD_CONGESTED
- POD_CONGESTION_L2
- POD_CONGESTION_L1
- POD_DANGER_OF_CONGESTION
- POD_PENDING_REQUEST_CONGESTED
- POD_PENDING_REQUEST_CONGESTION_L2
- POD_PENDING_REQUEST_CONGESTION_L1
- POD_PENDING_REQUEST_DANGER_OF_CONGESTION
- POD_CPU_CONGESTED
- POD_CPU_CONGESTION_L2

- POD_CPU_CONGESTION_L1
- POD_CPU_DANGER_OF_CONGESTION

For more information about alerts, see Common Alerts.

Maintain

Error logs are generated when the system is congested and the actions are taken to bring the system back to normal. Warning logs are generated to indicate the congestion level. However, error logs are not generated when messages are rejected to avoid additional resource usage to write error logs.

If you encounter alerts at system or application levels, see Alerts section for resolution steps.

In case the alerts still persist, perform the following:

- Collect the logs: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide*.
- Raise a service request: See My Oracle Support.

4.10 PDS Pod Congestion Control

The Policy Data Source (PDS) stores the User or Subscriber State Variable (SSV) information. PDS interacts with other Policy services, such as:

- The core services such as AM/UE/SM and PCRF Core services send requests to PDS. It would be a request to either fetch, update, or delete the SSV data.
- User service, SOAP Connector, and Diameter Connector send notification requests to PDS.
- Query service sends request to PDS to either get or delete SSV data. It also gets the default workflow details from the PDS.
- Audit service sends audit notification request to PDS.

At times, an excessive traffic from these services toward PDS service can be observed in the network, which can result in a high CPU utilization, high memory utilization. This can cause performance degradation in PDS service responses and eventually reach a state of service unavailability. Congestion control helps to identify such conditions and invoke load shedding rules to address these situations when these load conditions persist.

The PDS pod congestion control mechanism ensures consitent service availability to its consumer.

The pod congestion control mechanism involves:

1. Determining Pod Congestion State
2. Triggering Pod Congestion Control

Determining Pod Congestion State

In the pod congestion control mechanism, each PDS service pod monitors its congestion state. The congestion control works at 5 levels or states:

- NORMAL
- DANGER_OF_CONGESTION (DOC)
- CONGESTION_L1
- CONGESTION_L2

- CONGESTED

The pod's congestion state is decided based on CPU consumption and Queue.

1. **CPU**: The CPU usage for congestion state is calculated by comparing the CPU usage of the container (monitored using cgroup parameter, `cpuacct.usage`, which provides current cpu usage in nanoseconds) with the configured threshold.
2. **Queue**: For the DOC, CONGESTION_L1, CONGESTION_L2, and CONGESTED pod states, compare the number of pending messages in the queue with the configured pending messages threshold.

The PDS service pod's congestion states and their default congestion parameters, CPU, and Queue counts are provided in the following table:

Table 4-5 PDS Service Congestion States

Congestion States	CPU (%)	Queue Count (Pending Requests)
DANGER_OF_CONGESTION (DOC)	65	50
CONGESTION_L1	70	100
CONGESTION_L2	75	150
CONGESTED	80	200

To avoid toggling between these states due to traffic pattern, it is required for the pod to be in a particular state for a given period before transitioning to another state. The below configurations are used to define the period that the pod has to be in a particular state for:

- `stateChangeSampleCount`: This REST API parameter can be configured to specify after how many continuous intervals, the pod state can be changed. This value can range from 1 to 2147483647.
- `stateCalculationInterval`: This REST API parameter can be configured to specify the time duration or interval, after which the pod congestion state will be re-verified. This interval is configured in milliseconds and can range from 50 to 2147483647.

Triggering Pod Congestion Control

Every time PDS service receives requests from other services, it checks for the current congestion state of the pod. The Congestion Control mechanism is triggered if the pod's congestion state is in DOC or Congested_L1 or Congested_L2 or Congested.

The requests to the PDS service might have priority included as `oc-message-priority` attribute in the request header. The priority value ranges between 0 to 100 with 0 being the highest and 100 being the lowest priority.

> **Note**
>
> Currently, the downstream services do not propagate the `oc-message-priority` header to PDS service and will be implemented in future releases.

Priority-Based Load Shedding

Based on the pod's current congestion state, a load shedding rule is applied to perform priority-based load shedding. The load shedding rule is based on message priority. For example, when

the PDS service pod state is CONGESTED and the priority of discard messages is 30, then it determines if the message with the assigned priority should be rejected or accepted.

These rules get configured per congestion state. If there are no rules configured for a congestion state, then PDS service accepts the request as a default behavior. The user can configure the result codes for the rejected requests when configuring the load rules. The default result code is `503 Service Unavailable`.

The default load shedding rules for PDS service:

```
- state: DANGER_OF_CONGESTION
      discardPriority: 28
- state: CONGESTION_L1
      discardPriority: 20
- state: CONGESTION_L2
      discardPriority: 17
- state: CONGESTED
      discardPriority: 15
```

When PDS service is in congestion state, its response can be configured using PDS service advanced settings in CNC Console, using the key CONGESTION_RESPONSE_CODE. This key is used to configure the response code of the messages that are rejected by the PDS service due to pod's congestion state. By default, PDS service responds with a response code of 503. The response code configured should be 5xx error status only. Following is the list of configurable keys that can be added to set the message priority:

Table 4-6 Configuring Message Priority

Key	Default Value	Allowed Values
PDS_NOTIFY_USER_DATA_REQUEST_PRIORITY	18	0-100
PDS_GET_DEFAULT_WORKFLOW_REQUEST_PRIORITY	30	0-100
PDS_GET_USER_DATA_REQUEST_PRIORITY	18	0-100
PDS_UPDATE_USER_DATA_REQUEST_PRIORITY	18	0-100
PDS_DELETE_USER_DATA_REQUEST_PRIORITY	16	0-100
PDS_AUDIT_NOTIFY_REQUEST_PRIORITY	27	0-100
CONGESTION_RESPONSE_CODE	503	5xx

> **Note**
>
> For Policy application with Aspen Serivce Mesh (ASM) setups, add the `CONGESTION_RESPONSE_CODE` set to 500 using PDS Advanced Settings.

Example for Congestion Control Call Flow/Scenario

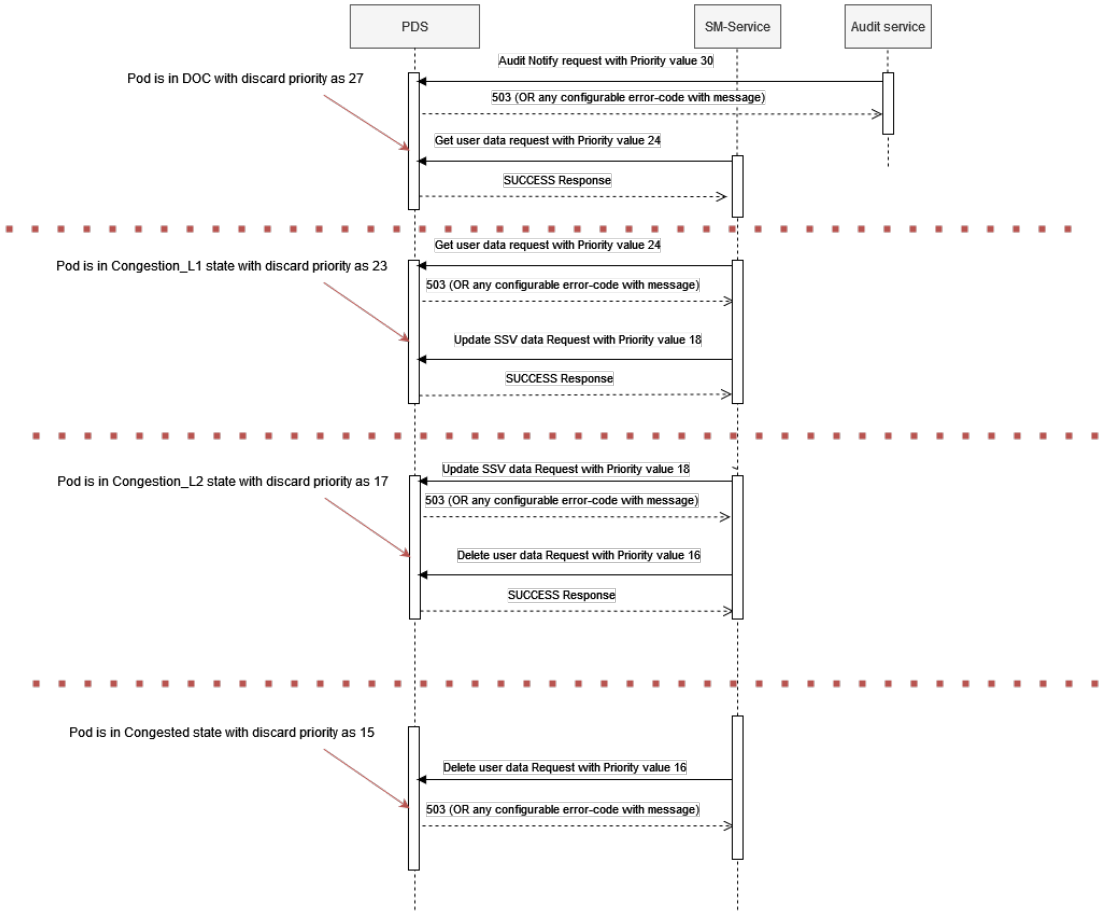
For the below call flows, consider the following load shedding rule configuration:

Table 4-7 Load Shedding Rule

Congestion State	Discard Priority
DOC	27
CONGESTION_L1	23
CONGESTION_L2	17
CONGESTED	15

The call flow describes the PDS service congestion control mechanism handling the requests from the other Policy services such as SM and Audit services when it is in different congestion state.

Figure 4-5 PDS Congestion Control Call Flow



PDS pod congestion is in DOC state and the discard priority is configured as 27:

- Audit service sends a notification request with discard priority value set to 30 toward PDS service.
- PDS service responds with a reject 503 status or with any configured error code and message.
- SM service sends a GET user information request with discard priority value set to 24 toward PDS service.

- PDS service responds with the success status.

PDS pod congestion is in Congestion_L1 state and the discard priority is configured as 23:

- SM service sends a GET user information request with discard priority value set to 24 toward PDS service.
- PDS service responds with a reject 503 status or with any configured error code and message.
- SM service sends an Update SSV request with discard priority value set to 18 toward PDS service.
- PDS service responds with the success status.

PDS pod congestion is in Congestion_L2 state and the discard priority is configured as 17:

- SM service sends an Update SSV request with discard priority value set to 18 toward PDS service.
- PDS service responds with a reject 503 status or with any configured error code and message.
- SM service sends a Delete user information request with discard priority value set to 16 toward PDS service.
- PDS service responds with the success status.

PDS pod congestion is in Congested state and the discard priority is configured as 15:

- SM service sends a Delete user information request with discard priority value set to 16 toward PDS service.
- PDS responds with a reject 503 or with any configured error code and message.

Managing PDS service Pod Congestion Control

Enable

By default, the Pod Congestion Control is disabled for PDS service. You can enable this feature using CNC Console or REST API for Policy.

Configure Using CNC Console

To enable the feature using CNC Console, set the **Enable** parameter in **Settings** page under **Congestion Control** for **Overload and Congestion Control Configurations**.

To configure SM service pod congestion control feature in CNC Console, see Congestion Control section.

To configure SM service pod congestion control discard messages in CNC Console, see the Advanced Settings section in PDS Settings section.

Configure Using REST API

Perform the feature configurations as described in "Congestion Control" section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Observability

Metrics:

The following metrics are used to provide information about this feature:

- `occnp_pod_congestion_state`

- occnp_pod_resource_stress
- occnp_pod_resource_congestion_state
- pod_cong_state_report_total
- pod_resource_congestion_state_report_total
- http_congestion_message_reject_total

For information about the metrics, see Pod Congestion Metrics.

Alerts

The following alerts are generated for this feature:

- POD_CONGESTED
- POD_CONGESTION_L2
- POD_CONGESTION_L1
- POD_DANGER_OF_CONGESTION
- POD_PENDING_REQUEST_CONGESTED
- POD_PENDING_REQUEST_CONGESTION_L2
- POD_PENDING_REQUEST_CONGESTION_L1
- POD_PENDING_REQUEST_DANGER_OF_CONGESTION
- POD_CPU_CONGESTED
- POD_CPU_CONGESTION_L2
- POD_CPU_CONGESTION_L1
- POD_CPU_DANGER_OF_CONGESTION

For more information about alerts, see Common Alerts.

Maintain

Warning logs are generated to indicate the congestion level. However, error logs are not generated when messages are rejected to avoid additional resource usage to write error logs.

If you encounter alerts at system or application levels, see Alerts section for resolution steps.

In case the alerts still persist, perform the following:

- Collect the logs: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide*.
- Raise a service request: See My Oracle Support.

4.11 OAuth Access Token Based Authorization

OAuth (Open Authorization) 2.0 an authorization protocol, allows a application to access resources hosted by other API applications on behalf of a user. OAuth provides an authentication layer and separates the role of client from that of the resource owner. On client requests access to resources controlled by the resource owner and hosted by the resource server, OAuth issues **Access Tokens**, a different set of credentials than those of the resource owner. An Access Token is a string that represents the authorization to access resources on behalf of the end user.

Policy supports OAuth 2.0 to authorize requests coming from consumer NFs. Policy application issues the Access Token as JSON Web Token (JWT). This enables Policy, the token issuer to include authorization data in the token itself. The consumer NF requests for Access Token from the issuer Policy, and sends them as part of its requests to Policy. Policy validates the requests and either approves or discards based on Access Token authorization received in the request.

The Access Token is validated with the configured public key certificate in Policy. Policy uses NRF Instance ID to validate the Access Token, where Ingress Gateway stored public keys against NRF instance Id. Policy also uses multiple public certificates for validating Access Tokens by adding support for Key-ID (K-ID) based access token validation, in addition to the existing NRF Instance ID based access token validation.

Ingress Gateway operates in the following three different OAuth Validation Modes:

- INSTANCEID_ONLY : This is the default OAuth Validation Mode used by Ingress Gateway. Ingress Gateway validates access token based on public keys indexed with NRF Instance ID in the issuer field.
- K-ID based ONLY: Ingress Gateway validates Access Token based on public keys indexed with key-id only.
- KID_PREFERRED (K-ID based with Instance ID based as fallback):
  - Ingress Gateway validates Access Token based on public keys indexed with Key-ID. If Key-ID is not FOUND in Access Token, Ingress Gateway attempts token validation using public keys indexed with NRF instance ID in the issuer field.
  - Fallback happens only if the received access token does not contain Key-ID or contains Key-ID but without public key configurations.

> **Note**
>
> - For more information on OAuth client functionality, see "OAuth Authentication and Validation" section in *Oracle Communications Cloud Native Core, Egress Gateway User Guide*.
> - For more information on OAuth validator functionality, see "Ingress Gateway Open Authorization (OAuth) Module" section in *Oracle Communications Cloud Native Core, Ingress Gateway User Guide*.
> - For more information on parameters and their supported values in Policy, see **OAUTH Configuration** section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*.
> - For information about OAuth access token attributes like kid, typ, iss, aud, scope etc., see *https://www.rfc-editor.org/rfc/rfc7515.html* page.

Managing OAuth Access Token Based Authorization

Token based Authorization Using Key-ID and NRF Instance ID

For configurations required to enable Access Tokens before deploying Policy, see "Configuring Secrets to Enable Access Token " section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide* .

Configure

To enable access token validation, configure both Helm-based and REST-based configurations on Ingress Gateway.

Configure Using Helm Configuration

For Helm configurations, see "OAUTH Configuration" section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide* .

REST API Configuration

After the Helm configuration, send the REST requests to use configured public key certificates. For more information about REST API configuration, see "OAuth Validator REST API Configuration" section in *Oracle Communications Cloud Native Core, Converged Policy REST API Specification Guide*.

Observability

Metrics

For information on Metrics, see "Oauth Metrics" section in Egress Gateway Metrics.

4.12 Support for Client Credentials Assertion (CCA) Header

The Client Credentials Assertion (CCA) is a security token that a client (typically a confidential client, like a web server or a backend service) presents to an authorization server in order to authenticate itself and obtain an access token. This assertion is commenly used in OAuth 2.0 for machine-to-machine communication, where the client needs to prove its identity without involving user interaction.

Client Credentials assertion (CCA) is a token signed by the Consumer NF. The 3GPP introduced Client Credentials Assertion (CCA) enables the producer NFs to authenticate the consumer NFs when using indirect communication through SCP. The CCAs are JSON web tokens and are secured with digital signatures based on JSON Web Signature (JWS). The consumer NF adds the 3gpp-Sbi-Client-Credentials header containing the CCA. The CCA contains the following:

- The NF instance ID of the consumer NF (subject)
- A timestamp (iat) and an expiration time (exp)
- The NF type of the expected audience, such as the NRF type or the NF type of the producer NF

The consumer NF digitally signs the generated CCA based on its private key. The signed CCA contains one of the following fields:

- The X.509 URL (x5u) to refer to a resource for the X.509 public key certificate or certificate chain used for signing the client authentication assertion
- The X.509 Certificate Chain (x5c) includes the X.509 public key certificate or certificate chain used for signing the client authentication assertion

Table 4-8 CCA Contents

Claim	Description
audience (aud)	NF type of the producer NF whose services are requested and/or NRF
expiration time (exp)	Timestamp of CCA expirarion
subject (sub)	NF instance ID of consumer NF
timestamp (iat)	Timestamp of CCA generation

Table 4-8 (Cont.) CCA Contents

Claim	Description
Security details used for signing CCA	• x5u - X.509 URL that refers to a resource for the X.509 public key certificate or certificate chain • x5c - X.509 Certificate Chain includes the X.509 public key certificate or certificate chain

The Producer NF, performs the CCA verification. It verifies that the public key certificate used to sign the CCA contains the same NF instance ID as the consumer's NF instance ID in the CCA. If the validation is successful, the Access Token request is processed further.

If the verification of the CCA fails at the receiving entity, a "403" response is returned with the cause attribute set to `CCA_VERIFICATION_FAILURE`.

If some rouge consumer NF acquires a valid CCA from some a consumer NF and uses it as its own CCA, then the CCA validation by the Producer NF creates a security gap. Therefore, additional CCA validations at Policy are needed to mitigate this security impact. The NF instance ID of the NFs in the CCA must match the NF instance ID in the certificate used in the TLS connection by the consumer NF as part of the Policy validations.

Managing Support for Client Credentials Assertion (CCA)

Using REST API: Perform the feature configurations as described in "CCA Header Validation" section in *Oracle Communications Cloud Native Core, Converged Policy REST API Specification Guide*.

4.13 Support for End-to-End Log Identifier Across Policy Services

Without a unique end-to-end log identifier, it is difficult to debug use cases where the problem can be at any microservice level.

This feature includes a unique end-to-end identifier to every log message, which can be used to identify the set of logs belonging to a given call flow across all the microservices. For example, a end-to-end log identifier is used across all microservices in an SM Create call flow. Similarly, end-to-end log identifiers used in an SM Update, SM Delete, SM Notify call flows across microservices involved in the flow. This end-to-end log identifier called ocLogId is included in the metadata of every log. the same identifier is used across all the microservices so that logs can be mapped between microservices. Also, a common identifier is used to track a user per microservice.

This feature supports the following call flow for Policy microservices:

- SM Create
- SM Update
- SM Delete
- SM Notify
- SM Rx call flow
- SM Rx RAR call flow
- SM Rx STR call flow

- AM Create
- AM Update
- AM Delete
- AM Notify
- UE Create
- UE Update
- UE Delete
- UE Notify
- AMF discovery for UE Create
- AMF discovery for UE Update
- NRF discovery for Alternate Route Retry call flow
- Binding service call flow for stale session detection
- Audit call flow using Audit service

Separate log identifiers are created in every georedundancy site. That is, every site creates its own identifier and log the same.

Note

Currently, only Ingress Gateway and Diameter Gateway services can generate ocLogId (when a request from an external NF arrives) and Audit when sending a notification. As a result, there are certain flows and services that are not part of the reception, addition to log metadata and propagation of ocLogId and for which Log Correlation is not ensured. Some of these unsupported services and flows worth to mention are the following:

The following are some of those services for which this feature is not supported:

- Configuration Flows: Configuration flows for each service are not supported as these involve only the services for which configuration is changed and the services where the configuration was added through.
- Config Service: Config service is not supported as the configuration is either changed or the configuration is add through.
- Base Diameter Messages as CER/A and DWR/A. ocLogId will not be generated or propagated.
- CMService: Log Correlation is not supported and CMservice can not generate ocLogId. Any flow that was originated by CMService is not supported, including configuration and session viewer flows.
- Query Service: Query service is not supported. This service can not receive or propagate ocLogId.
- LDAP-Gateway
- Alternate Route Service

If any of the supported services receive requests originated or coming from any of the not yet supported ones, directly or indirectly, ocLogId is not ensured to be received as the propagation through any valid mechanism might be yet implemented.

End-to-end log identifier at different services

The Ingress Gateway creates the ocLogId and adds the same in each log that is generated.

The Ingress Gateway passes the ocLogId in the header to Policy microservice such as SM service.

> **Note**
>
> For SM Rx call flows (AAR and STR), the ocLogId is generated by Diameter Gateway instead of Ingress gateway.
>
> When Audit service sends a notification to the other services, the ocLogId is generated by Audit service

Logging by SM service

- When SM service receives the ocLogId from Ingress Gateway, it uses the ocLogId while generating the logs. This information is included as part of all the log levels.
- SM service in turn passes the identifier to other backend services such as PRE, PDS, Egress Gateway, Bulwark, and Binding service.
- When SM service receives the ocLogId for the notification from Audit service, they log the messages with ocLogId.
- When SM service receives the ocLogId for the notification from PDS Service or Binding Serivce, they log the messages with ocLogId.
- When SM service receives the ocLogId from Diameter Connector for Rx call flow, it uses the ocLogId while generating the logs. This information is included as part of all the log levels.
- SM service in turn passes the identifier as header to other backend services.

Logging by AM service

- When AM service receives the ocLogId from Ingress Gateway, it uses the ocLogId while generating the logs. This information is included as part of all the log levels.
- AM service in turn passes the identifier to other backend services such as PRE, PDS, Egress Gateway, and NRF Client.
- When AM service receives the ocLogId for the notification from Audit service, they log the messages with ocLogId.
- When AM service receives the ocLogId for the notification from PDS service or Binding Serivce, it logs the messages with ocLogId.

Logging by UE Policy service

- When UE Policy service receives the ocLogId from Ingress Gateway, it uses the ocLogId while generating the logs. This information is included as part of all the log levels.
- UE Policy service in turn passes the identifier to other backend services such as PRE, PDS, Egress Gateway, and NRF Client.
- When UE Policy service receives the ocLogId for the notification from Audit service, they log the messages with ocLogId.

- When UE Policy service receives the ocLogId for the notification from PDS service or Binding Serivce, it logs the messages with ocLogId.

Logging at PDS

- When PDS receives the ocLogId from SM service, AM service, or UE Policy service, it uses the ocLogId while generating the logs.
- PDS passes this ocLogId as the header to the backend services such as UDR Connector, CHF Connector, LDAP, Diameter Gateway, Diameter Connector.
- When PDS receives the ocLogId for the notification from UDR Connector, CHF Connector, Diameter Gateway, Diameter Connector, or Audit service, PDS logs the messages with ocLogId.
- PDS passes the ocLogId as the header to Policy services such as SM service, AM service, or UE Policy service, while sending the notification. PDS uses the ocLogId saved in the context and does not generate any new identifier.

> **Note**
>
> Only when PDS receives a notification from CHF or UDR, a new ocLogId gets created.

Logging at UDR Connector

- When UDR Connector receives the ocLogId from PDS, it uses the ocLogId while generating the logs.
- UDR Connector passes this ocLogId as the header to the backend services such as Egress Gateway.
- When UDR Connector receives the ocLogId for the notification from Ingress Gateway, it logs the messages with ocLogId.

Logging at CHF Connector

- When CHF Connector receives the ocLogId from PDS, it uses the ocLogId while generating the logs.
- CHF Connector passes the ocLogId as the header to the backend service such as Egress Gateway.
- When CHF Connector receives the ocLogId for the notification from Ingress Gateway, it logs the messages with ocLogId.

Logging at Egress Gateway

When Egress Gateway receives the ocLogId from SM service, CHF Connector, UDR Connector, AM service, or UE Policy service, it uses the ocLogId while generating the logs.

Logging at Audit service

- When Audit service detects a possible stale session, audit service generates the ocLogId.
- Audit service includes the ocLogId in the header and sends it to SM service, AM service, UE Policy service, PDS, and Binding service.

Logging at Binding service

- When Binding service receives the ocLogId for the SM service or Audit service, it logs the messages with ocLogId.

- Also, Binding service receives the ocLogId when BSF sends a request to check if a PcfBinding is still valid or not.

Logging at Diameter Gateway

- Diameter Gateway generates the ocLogId when it receives a AAR (AAR-i or AAR-U) and also when it receives a STR message.
- Diameter Gateway uses the ocLogId while generating the logs.
- Diameter Gateway sends the ocLogId as headers/AVP to the backend services such as Diameter Connector and Binding service.

Logging at Diameter Connector

- When Diameter Connector receives the ocLogId from Diameter Gateway, it logs the messages with ocLogId.
- Diameter Connector sends the ocLogId as header to the backend services such as SM service.

Logging at PRE

- When PRE receives the ocLogID from SM service, AM service, UE Policy service, or PDS, it uses the ocLogId while generating the logs.
- PRE sends the ocLogId as headers/AVP to the backend service.

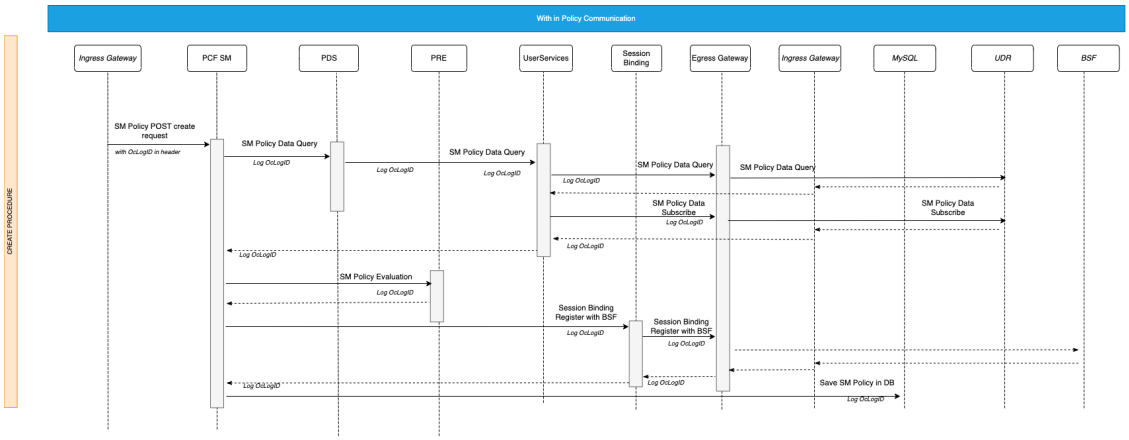
Logging at Bulwark service

When Bulwark service receives the ocLogID from SM service, it uses the ocLogId while generating the logs.

Logging at NRF Client

- When NRF Client receives the ocLogId from AM service or UE Policy service, it logs the messages with ocLogId.
- NRF Client sends the ocLogId as headers to the backend services.
- When there is a request initiated by NRF Client (not something from UE or AM) NRF creates the ocLogId by itself and sends it to the subsequent services.

Figure 4-6 Example: Unique Identifier for SM Create Call Flow



1. Ingress Gatway receives SM session create request from SMF. Ingress Gateway generates the ocLogId and passes the ocLogId in the header to SM service.

2. SM service sends a Get request to PDS to get the SMAssociation details and uses the ocLogId while generating the logs.
3. PDS sends the query to User Service (UDR Connector, CHF Connector). PDS uses the ocLogId while generating the logs. Also, PDS passes this ocLogId as the header to the backend services such as UDR Connector and CHF Connector.
4. UDR Connector inturn forwards the request to UDR through Egress Gateway. UDR Connector uses the ocLogId while generating the logs. Also, it passes the ocLogId as a header to Egress Gateway.
5. UDR responds to the request with the SM Policy data.
6. User service then sends a request to UDR for the subscription through Egress Gateway. The logs generated at Egress Gateway are saved with ocLogId.
7. UDR responds with the subscription details. UDR Connector forwards the subscription details to SM service and logs the transaction with ocLogId.
8. SM service interacts with PRE for Policy evaluation. PRE uses the ocLogId while generating the logs.
9. SM service sends a session binding request with BSF and includes the ocLogId in the header. The session binding request containing ocLodId is sent to BSF through Egress Gateway.
10. SM service receives the response to session binding request from BSF through Ingress Gateway. SM service updates the details in SMAssociation database and logs the transaction with ocLogId.

Managing the End-to-End Log Identifier across Policy Services

`TRACE_ID_GENERATION_ENABLED` Helm parameter is used to enable or disable the addition of the end-to-end identifier through Ingress Gateway. This same flag is also present in Diamater Gateway, Audit service, and NRF Client.

By default, `TRACE_ID_GENERATION_ENABLED` is enabled.

Audit service uses it when initiating a stale session notification.

Diameter Gateway uses it when receiving an AAR and STR messages.

NRF Client uses it when initiating a request by its own (a request that was not received from other services).

By default, the value this parameter is set to true and the feature is enabled.

When this feature is enabled, Ingress Gateway generates the ocLogId and propagates the headers to the succeeding microservices, Audit service, Diameter Gateway, NRF Client.

Ingress Gateway passes the ocLogId generated as the header and sends the same to other Policy microservices such sa SM service, AM service, or UE Policy service.

4.14 Support for Automated Certificate Lifecycle Management

Public Key Interface (PKI) is the set of elements such as public/private keys, certificate signing request, and certificates that are required to handle secure communications and transactions. Policy uses secure protocols for its communications, such as HTTPS and Secure Socket Layer (SSL) / Transport Layer Security (TLS) technologies to handle these secure communications. This is achieved with the use of Public and Private Keys, and the presence of trusted authorities, also known as Certificate Authorities (CA), which create and issue certificates. These certificates have a determined validity period. These certificates must be renewed

before expiry. They can also be revoked when the CA or its keys are compromised. These certificates must be recreated when required.

This feature enables Policy to support automation of certificate lifecycle management in integration with Oracle Communications Certificate Manager (OCCM).

OCCM provides the option to automatically create, renew, and delete certificates for a given CA, with the possibility to track previously created certificates and renew/delete them when required.

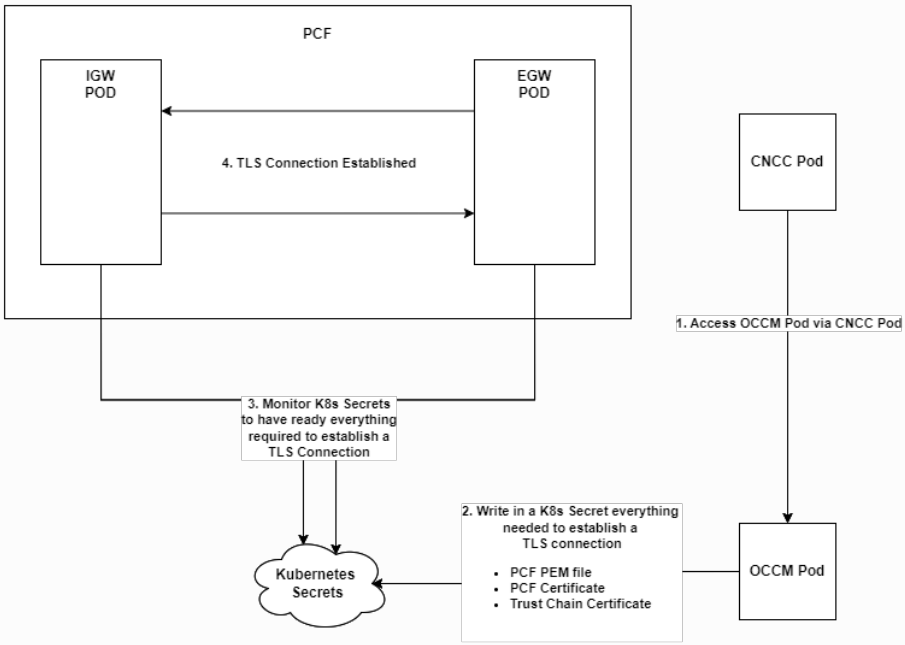
The certificate lifecycle management includes:

- Certificate Creation,
- Certificate Deletion,
- Certificate Monitoring (including the ones that were created using a different tool from OCCM),
- Certificate Renewal.

Note

OCCM do not support OAuth token generation and distribution (currently handled by NRF) used for SBI signalling.

Figure 4-7 Policy Integration with OCCM



There is no direct communication between OCCM and Policy. All the communications are handled using monitoring Kubernetes Secrets.

All the required certificates are configured using OCCM.

After OCCM creates these Kubernetes Secrets, or monitors the already existing ones, the Ingress and Egress Gateways monitor these Secrets and keep track of their current status:

- **VALID**: A Kubernetes Secret which holds a certificate that has not expired and it is properly signed,
- **EXPIRED**: A Kubernetes Secret which holds a certificate that has met its expiration date (the value determined in its `notAfter` value),
- **MISSING**: A Kubernetes Secret which has its certificate missing, or any other essential file for the TLS/SSL bundle.
- **CORRUPT**: A Kubernetes Secret which has its certificate corrupt, either invalid file, invalid signature, or invalid format.

Managing the keys and certificates

Install Guide Considerations

- Upgrade: When Policy is deployed with OCCM, follow the specific upgrade sequence as mentioned in the *Oracle Communications, Cloud Native Core Solution Upgrade Guide*.
- Rollback: You can remove Kubernetes secrets if the current version of Policy does not use that secret by checking the `occnp_custom_values.yaml` file. Before deleting, please make sure that there is no plan to rollback to the Policy version which uses these secrets. Otherwise Rollback will fail. For more information on migrating the secrets from Policy to OCCM and removal of Kubernetes secrets from the yaml file, see *Upgrade Strategy* in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.

Configure

To configure HTTPS in ingress-gateway, the following parameters must be configured in custom-value.yaml file in the ingress-gateway section:

- ingress-gateway.enableIncomingHttps
- ingress-gateway.service.ssl.privateKey.k8SecretName
- ingress-gateway.service.ssl.privateKey.k8NameSpace
- ingress-gateway.service.ssl.privateKey.rsa.fileName
- ingress-gateway.service.ssl.certificate.k8SecretName
- ingress-gateway.service.ssl.certificate.k8NameSpace
- ingress-gateway.service.ssl.certificate.rsa.fileName
- ingress-gateway.service.ssl.caBundle.k8SecretName
- ingress-gateway.service.ssl.caBundle.k8NameSpace
- ingress-gateway.service.ssl.caBundle.fileName
- ingress-gateway.service.ssl.keyStorePassword.k8SecretName
- ingress-gateway.service.ssl.keyStorePassword.k8NameSpace
- ingress-gateway.service.ssl.keyStorePassword.fileName
- ingress-gateway.service.ssl.trustStorePassword.k8SecretName
- ingress-gateway.service.ssl.trustStorePassword.k8NameSpace
- ingress-gateway.service.ssl.trustStorePassword.fileName

For more information, see *Basic Configurations in Ingress Gateway* section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.

To configure HTTPS in egress-gateway, configure the following parameters under *egress-gateway* section in *custom-value.yaml* file:

- egress-gateway.enableOutgoingHttps
- egress-gateway.egressGwCertReloadEnabled
- egress-gateway.egressGwCertReloadPath
- egress-gateway.service.ssl.privateKey.k8SecretName
- egress-gateway.service.ssl.privateKey.k8NameSpace
- egress-gateway.service.ssl.privateKey.rsa.fileName
- egress-gateway.service.ssl.privateKey.ecdsa.fileName
- egress-gateway.service.ssl.certificate.k8SecretName
- egress-gateway.service.ssl.certificate.k8NameSpace
- egress-gateway.service.ssl.certificate.rsa.fileName
- egress-gateway.service.ssl.certificate.ecdsa.fileName
- egress-gateway.service.ssl.caBundle.k8SecretName
- egress-gateway.service.ssl.caBundle.k8NameSpace
- egress-gateway.service.ssl.caBundle.fileName
- egress-gateway.service.ssl.keyStorePassword.k8SecretName
- egress-gateway.service.ssl.keyStorePassword.k8NameSpace
- egress-gateway.service.ssl.keyStorePassword.fileName
- egress-gateway.service.ssl.trustStorePassword.k8SecretName
- egress-gateway.service.ssl.trustStorePassword.k8NameSpace
- egress-gateway.service.ssl.trustStorePassword.fileName

For more information, see *Basic Configurations in Egress Gateway* section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.

Observability

Monitoring the keys and certificates

Policy supports monitoring and automatic renewal of its' TLS certificates in integration with OCCM.

It is validated that the renewed certificate and key are picked up for any new TLS connections.

Also, the existing TLS connections using the previous key and certificate are gracefully brought down.

Clean up of the certificates are also handled through OCCM.

For information about enabling HTTPS, see *Configuring Secrets for Enabling HTTPS* in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.

.

Metrics

The oc_certificatemanagement_tls_certificate_info metric is used to support automated certificate lifecycle management.

For more information, see Metrics for Automated Certificate Lifecycle Management.

4.15 Support for Adding Reduced Capability to UEs

Policy supports handling the requests from reduced capability devices (RedCap) to support IoT ecosystem.

RedCap is a new device class that is designed to provide a cost-effective way to connect devices that do not require the full capabilities of the 5G system. It reduces UE complexity through fewer RX/TX antennas, reduced UE use of bandwidth, lower UE power consumption, relaxed data rates, relaxed UE processing time and processing capability to help enable some exciting use cases, primarily around Industrial wireless sensors, video surveillance and wearables.

Policy identifies the incoming request from the RedCap devices based on the NR_REDCAP enumerated value in the request that indicates NR RedCap access type.

Based on this value, Policy interacts with PRE to make appropriate decisions for the reduced capability UEs.

> **Note**
>
> Policy supports policy decisions on NR_RedCap rat_type for:
>
> - create, update, and update notify message for SM service
> - create message for AM service and UE Policy service

Managing the support for reduced capability devices

A new value `NR_REDCAP` is added to the existing **RatType** block under Policy Projects in CNC Console.

For more information on the RatType block, see *PCF UE Policy* section in *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.

4.16 PCF Support for ME-XX String in Server Header from CHF

When PCF communicates with other NFs, it makes use of a header called server, as defined in 3GPP. PCF uses these header is used by PCF to make decisions on retry strategies when receiving error responses. The header contains the instance or instances that generated the error. Based on the standard defined in 3GPP, the instance(s) inserted in the header must follow the following pattern: **"<NFType>-<NF Instance ID>"** for an NF, where

- **<NF Type>** is the type of the NF or network entity generating the error, set to the NFType value as defined in 3GPP.
- **<NF Instance ID>** is the identity of the NF or network entity generating the error, set to the FQDN of the SCP or SEPP, or to the NF Instance ID of the HTTP server.

For example, when inserted by an NF such as SCP or SEPP, the error originators, the server header pattern looks like "SCP-<SCP FQDN>" for a SCP and "SEPP-<SEPP FQDN>" for a SEPP.

The information carried in the Server Header is useful for troubleshooting. There might be cases where this server header has custom information and does not follow header pattern as defined by the 3GPP. Considering this, if a differently formatted value such as "ME-me-instanceid", is inserted in the header alongside the 3GPP supported ones, PCF, at present, completely ignores the server header, which impacts the retry strategy.

With this feature implementation, PCF core (SM/AM/UE) services ignores the custom server header value such as "ME-me-instanceid" and continue considering the valid SCP, SEPP or other NF type Server Headers.

Model C: PCF (SM/AM/UE) as Consumer Supports Custom Server Header from CHF on N28 Interface

The N28 reference point is defined for the interactions between PCF and Charging Function (CHF). Policy sends initial POST request for subscription to policy counter information from the CHF over N28 interface. If the CHF cannot successfully fulfill the received POST request due to an error, then CHF sends error response to PCF.

On receiving the error response from CHF, the PCF Core services check for `Server Header Support` field status,

- if the field status is set to `Single Instance`, then the regular expression evaluates the server header value. From the header value, only one NF instance of the primary/ secondary CHF Producer is filtered and custom server header content such as ME-me-instanceid is ignored.
- if the field status is set to `Multi Instance`, then the regular expression evaluates the server header value. From the header value. two NF instances from all the producer NF instances in the NFSet is filtered and custom server header content such as ME-me-instanceid is ignored.

While creating retry profile, the regular expression to ignore the content in server header is configured for the Policy services. In the CNC Console, a field **Ignore Custom server header value** is added in **Create Retry Profile** section under **Retry Profile**. In this field, you can specify the string pattern that can be ignored in the server header value.

The following table maps the configured regular expression that identifies the matching pattern in the custom server header value. The identified pattern, if exists, is ignored and the remaining value is considered as seen in the **Filtered Content** column in the following table.

Table 4-9 CHF Server Header Content Pattern Matching Regular Expression

Configured Regular Expression	Server Header Content	Filtered Content	Server Header Support
ME-.+	ME-me-54804518-4191-46b3-955cac631f953eb91 CHF-54804518-4191-46b3-955cac631f953eb91	CHF-54804518-4191-46b3-955cac631f953eb9	Single Instance

Table 4-9 (Cont.) CHF Server Header Content Pattern Matching Regular Expression

Configured Regular Expression	Server Header Content	Filtered Content	Server Header Support
ME-.+	ME-me-54804518-4191-46b3-955c-ac631f953ed82 CHF-54804518-4191-46b3-955c-ac631f953ed82 ME-me54804518-4191-46b3-955c-ac631f953eb91 CHF-54804518-4191-46b3-955c-ac631f953eb91	CHF-54804518-4191-46b3-955c-ac631f953ed82 CHF-54804518-4191-46b3-955c-ac631f953eb91	Multi Instance
ME-.+	CHF-54804518-4191-46b3-955cac631f953eb91 ME-me-54804518-4191-46b3-955cac631f953eb91	CHF-54804518-4191-46b3-955cac631f953eb91	Single Instance
ME-.+	CHF-54804518-4191-46b3-955c-ac631f953ed82 CHF-54804518-4191-46b3-955cac631f953eb91	CHF-54804518-4191-46b3-955c-ac631f953ed82 CHF-54804518-4191-46b3-955cac631f953eb91	Multi Instance
(ME|CHF)-.+	ME-me-54804518-4191-46b3-955cac631f953eb91 CHF-54804518-4191-46b3-955cac631f953eb91	Empty	Single Instance

Managing PCF Support for ME-XX String in Server Header from CHF

Enable

PCF supports ME-XX String in Server Header from CHF as a core functionality. You do not need to enable or disable this feature.

Configure Using CNC Console

To add the regular expression in CNC Console, set the **Ignore Custom server header value** field in **Create Retry Profile** section under **Retry Profile** for **Common Data Configurations** for Policy Services.

Configure Using REST API

Perform the feature configurations as described in "Retry Profile" section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Maintain

Error logs are generated when the system is congested and the actions are taken to bring the system back to normal. Warning logs are generated to indicate the congestion level. However, error logs are not generated when messages are rejected to avoid additional resource usage to write error logs.

If you encounter alerts at system or application levels, see Alerts section for resolution steps.

In case the alerts still persist, perform the following:

- Collect the logs: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide*.
- Raise a service request: See My Oracle Support.

4.17 Sy SLR Enhancements for Signalling Updates and UDR Notification

The Sy reference point is located between CnPolicy and Online Charging System (OCS). It enables transfer of policy counter status information relating to subscriber spending from OCS to CnPolicy.

CnPolicy considers the subscriber's spending status into account for its Policy decisions. The CnPolicy requests the spending limit report for policy counters from the OCS using the initial or intermediate spending limit request. Currently, CnPolicy supports fetching of OCS counters in the following scenario in a synchronous mode:

- In the initial request i.e., when the request is sent for the first time for a Subscriber, the cnPCRF sets the SL-Request-Type AVP to the value, INITIAL_REQUEST and initiates a diameter Sy Spending Limit Request (SLR) towards OCS.
- On a subsequent request on the second IP CAN session for the same subscriber, the PCRF sets the SL-Request-Type AVP to the value, INTERMEDIATE_REQUEST and updates the OCS counters.

With this feature, CnPCRF and CnPolicy fetches OCS counter status over Sy reference point in the following scenarios in a synchronous mode:

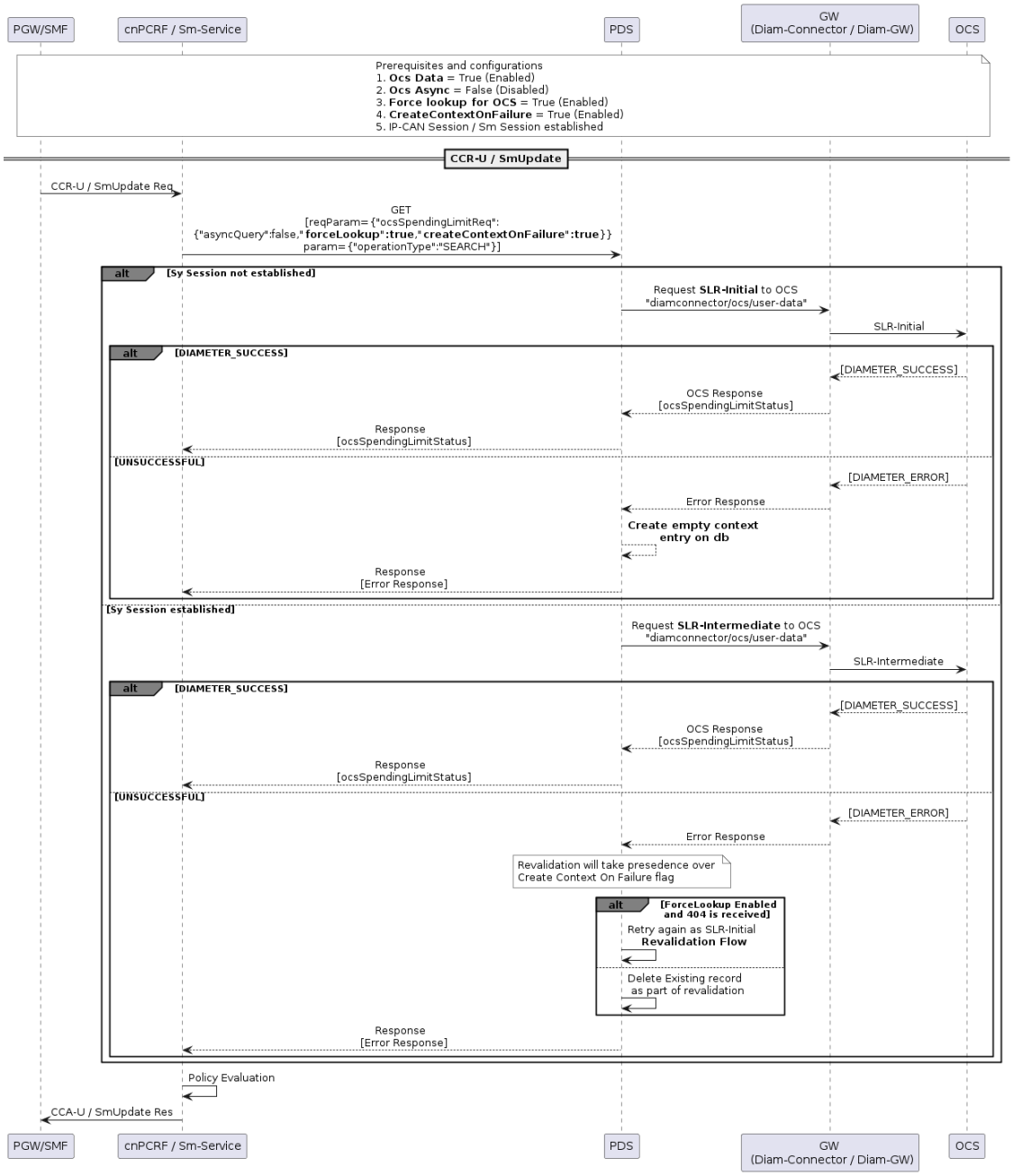
- CnPCRF initiates a Sy SLR intermediate request on receiving CCR-U or SM update request for a given subscriber session-id. CnPolicy evaluates policy based on the latest counters received over Sy and sends the policy rules in the CCA-U or SM update response.
- On receiving UDR notification, cnPCRF issues an Sy SLR initial or intermediate requests. CnPolicy evaluates policy and triggers a RAR or Update Notify toward PGW/SMF for all IP CAN or PDU sessions of the subscriber.

Update Signaling Flow for CnPCRF/SM Services

CnPCRF/SM service, on receiving a CCR update or SM update requests from PGW/SMF, performs OCS counters lookup by invoking OCS Sy Spending Limit Request in a synchronous mode to OCS. At PCRF Core and PCF SM services, a flag **Enable Force Lookup for OCS** is added. You can enable or disable this flag in the PCRF Core service or PCF Session Management **Settings** page under **User** tab in the **OCS Spending Limit** section in the CNC Console. By default, this flag is disabled. The purpose of this flag is to force PDS to perform OCS Lookup, but at PDS this will be treated as revalidation. If this flag is enabled in the core services, then while initiating the OCS spending limit request the "forceLookup" is sent as true toward PDS.

The following call flow describes the fetching of Sy OCS counters on receiving session updates requests from PGW/SMF:

Figure 4-8 Update signalling flow for CnPCRF/SM service



1. PGW/SMF sends a CCR Update or SM Update request to CnPCRF/SM service.
2. The CnPCRF/SM service sends a GET OCSSpendingLimitRequest with "forceLookup" set to true to PDS.
3. If Sy session is not established then PDS sends SLR-Initial request to Diameter Gateway.
4. Diameter Gateway forwards this request to OCS.
5. If OCS responds with DIAMETER_SUCCESS code to Diameter Gateway.
   a. Diameter Gateway responds with OCS response containing `ocsSpendingLimitStatus` information to PDS.

   b. PDS forwards the OCS response with `ocsSpendingLimitStatus` data to CnPCRF/SM service.

6. If OCS responds with DIAMETER_ERROR code to Diameter Gateway.
   a. Diameter Gateway sends the Error Response to PDS. At PDS, an empty context entry is created in the Database if CreateContextOnFailure is enabled from the request from cnPCRF/PCF-SM.
   b. PDS forwards the Error Response to CnPCRF/SM service.
7. If Sy session is already established then PDS sends SLR-Intermediate request to Diameter Gateway
8. Diameter Gateway forwards this request to OCS.
9. If OCS responds back with DIAMETER_SUCCESS code to Diameter Gateway.
   a. Diameter Gateway responds with OCS response containing `ocsSpendingLimitStatus` information to PDS.
   b. PDS forwards the OCS response with `ocsSpendingLimitStatus` data to CnPCRF/SM service.
10. If OCS responds back with DIAMETER_ERROR code to Diameter Gateway. Diameter Gateway sends the error response to PDS. At PDS,
    a. On receiving the response, it evaluates the flag `createContextOnFailure` in the cnPCRF/PCF SM request and if enabled the existing context is updated as dummy context in the database.
    b. With Force Lookup flag enabled, PCF forces PDS to do a lookup towards OCS, but PDS treats this flag as if was part of revalidation flow and retries again as SLR-Initial Revalidation flow.
    c. Deletes the existing record in the database as part of the revalidation procedure.
11. PDS forwards the Error Response to CnPCRF/SM service.
12. CnPCRF/SM service sends a request with OCS information to PRE for Policy evaluation.
13. Policy evaluated information is sent to PGW/SMF as CCA update or SM update response.

UDR Notification Call Flow

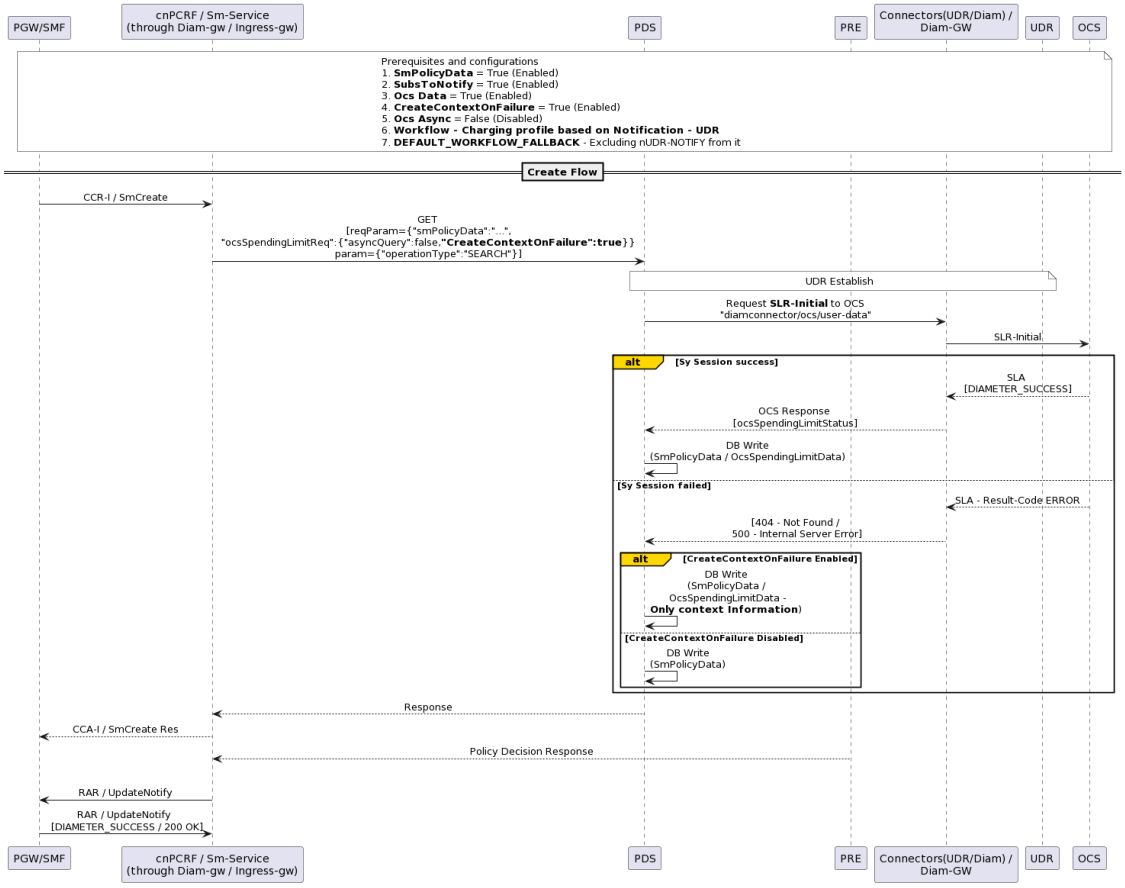
In the UDR notification call flow, the PDS workflow **Charging Profile selection based on Notification - UDR** is added to the list of the existing workflows. On selecting this workflow PDS is instructed to select the user profile based on the notification from UDR. The **Charging Profile selection based on Notification - UDR** workflow is based on the existing **Charging Profile selection based on User Profile - UDR** workflow with two extra tasks added to UDR notify workflow namely, the search and PRE tasks.

In order to use the UDR notification flow that initiates any lookup towards OCS, you need to have the following configurations to be set:

- Select the new workflow **Charging Profile selection based on Notification - UDR** type.
- PDS-OCS dummy context, that is created using either CCR-I/SmCreate or CCR-U/SmUpdate must exist in PDS Database.
- PDS-OCS sy session, that is created using either CCR-I/SmCreate or CCR-U/SmUpdate must exist in PDS Database.

The following call flow describes the fetching of Sy OCS counters during a CCR I/SM Create request from PGW/SMF:

Figure 4-9 UDR PDS - Create Flow

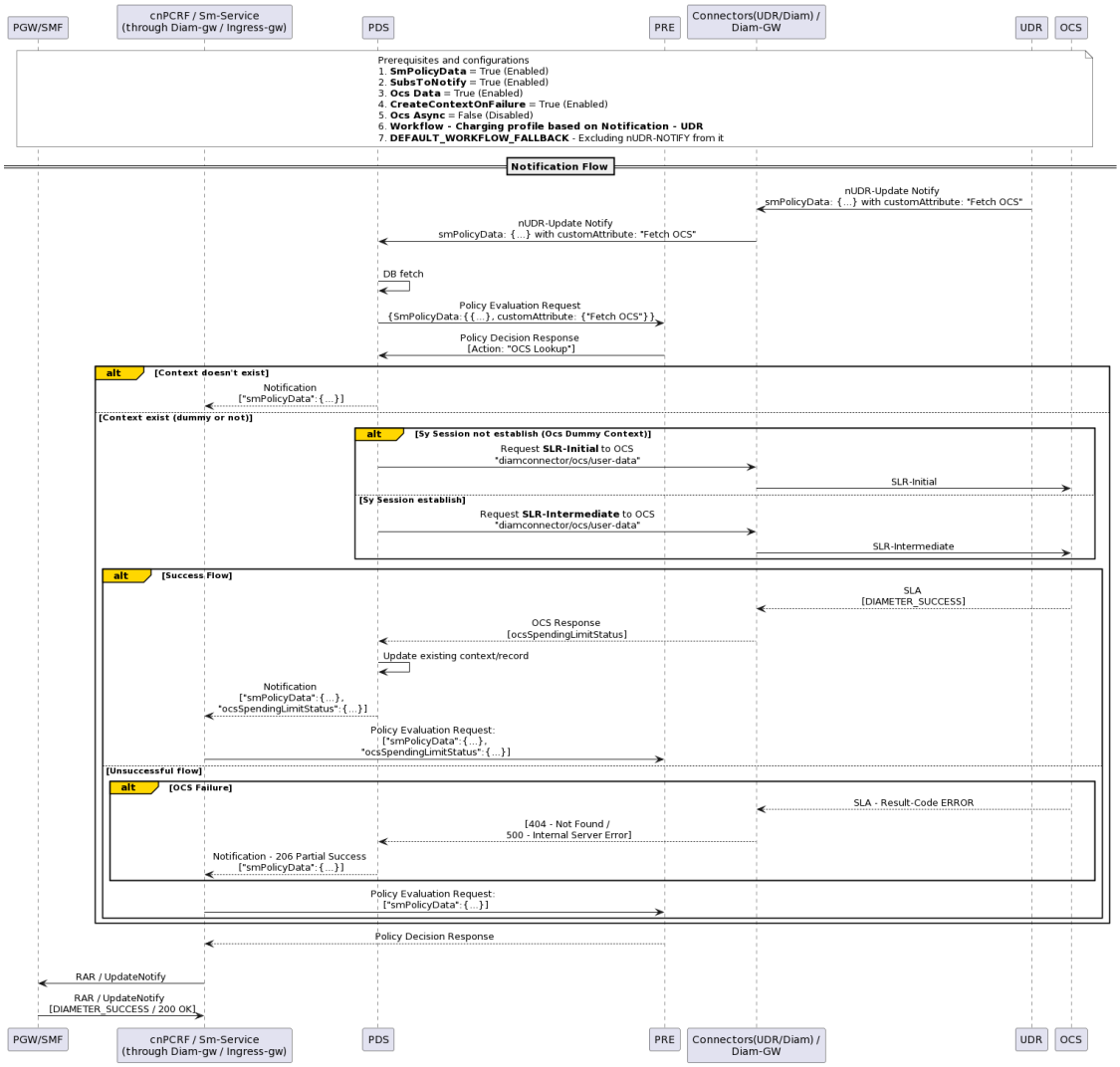


1. PGW/SMF sends a CCR I or SM Create request to CnPCRF/SM service.
2. The CnPCRF/SM service sends a GET OCSSpendingLimitRequest with "createContextonFailure" set to true to PDS.
3. PDS sends SLR-Initial request to Diameter Gateway.
4. Diameter Gateway forwards this request to OCS.
5. If Sy session creation is successful with OCS then it returns DIAMETER_SUCCESS code to Diameter Gateway.
   a. Diameter Gateway responds with OCS response containing `ocsSpendingLimitStatus` information to PDS.
   b. At PDS the `ocsSpendingLimitData` and `smPolicyData` is written into the Database.
6. If Sy session creation is unsuccessful with OCS then it returns DIAMETER_ERROR code to Diameter Gateway.
   a. Diameter Gateway responds with either `404 Not Found` or `500 Internal Server Error` to PDS.
   b. PDS checks for the attribute "createdContextOnFailure" value. On finding it enabled PDS writes only the OCS context information and `smPolicyData` into the Database.
   c. PDS checks for the attribute "createdContextOnFailure" value. On finding it disabled PDS writes only `smPolicyData` into the Database.

7. PDS responds to CnPCRF/SM service.
8. CnPCRF/SM service triggers a CCA I or SM Create response to PGW/SMF.
9. On receiving PRE decision, CnPCRF/SM service triggers a Gx RAR/SM update notify response to PGW/SMF.

The following call flow describes the fetching of Sy OCS counters during a UDR Notification:

Figure 4-10 UDR PDS Notification Flow



1. UDR triggers an UDR Update Notify request with customAttribute set to "FetchOCS" toward Diameter Gateway.
2. Diameter Gateway forwards this request to PDS.
3. PDS performs a DB fetch and sends Policy evaluation request with the UDR Notification information to PRE.
4. PRE responds back with Policy decision request with an action of "OCS Lookup" to PDS.
5. If OCS Context does not exist then PDS sends notification request to CnPCRF/SM service.

6. If either the dummy OCS Context or actual OCS Context exists then PDS initiates either Sy SLR Initial or Sy SLR Intermediate request to OCS via Diameter Gateway.
7. If OCS sends DIAMETER_SUCCESS code to Diameter Gateway then following call flow runs:
    a. Diameter Gateway responds with OCS response containing `ocsSpendingLimitStatus` information to PDS.
    b. PDS updates the existing OCS context in the database with received OCS data.
    c. PDS sends a notification request with `ocsSpendingLimitStatus` data to CnPCRF/SM service.
    d. CnPCRF/SM service sends a Policy evaluation request with `ocsSpendingLimitStatus` to PRE.
8. If OCS sends DIAMETER_ERROR code to Diameter Gateway then following call flow runs:
    a. Diameter Gateway responds with either `404 Not Found` or `500 Internal Server Error` to PDS.
    b. PDS sends notification with 206 Partial success to CnPCRF/SM service.
    c. CnPCRF/SM service sends a Policy evaluation request to PRE.
9. PRE sends Policy decision response to CnPCRF/SM service.
10. CnPCRF/SM service triggers a Gx RAR/SM update notify response to PGW/SMF.

Managing Sy SLR Enhancements for Signalling Updates and UDR Notification

Enable

By default, the Sy SLR Enhancements for Signalling Updates and UDR Notification feature is disabled. You can enable this feature using CNC Console or REST API for Policy.

Configure Using CNC Console

To configure the feature using CNC Console, set the **Enable force lookup on Update** parameter to true in **OCS Spending Limit** section under **user** tab on the **Settings** page for **PCRF Core Service Configuration**.

To enable "createContextOnFailure" during UDR Notification flow using the CNC Console, add following keys:

- "USER.ocsSpendingLimit.createContextOnFailure" parameter in the Advanced Settings key in PCRF Core Service Configurations Settings section.
- "USER.CREATE_CONTEXT_ON_FAILURE_OCS_DATA" parameter in the Advanced Settings key in PCF Session Management section.

In the CNC Console the PDS workflow **Charging Profile selection based on Notification - UDR** is selected at PDS Workflow section.

Configure Using REST API

Perform the feature configurations as described in "PCRF Core Service", "Session Management Service" and "PDS Workflow" sections in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Observability

Metrics:

The metric `create_context_on_failure_response` is added to Diameter Connector Service Metrics section.

Maintain

Error logs are generated when the system is congested and the actions are taken to bring the system back to normal. Warning logs are generated to indicate the congestion level. However, error logs are not generated when messages are rejected to avoid additional resource usage to write error logs.

If you encounter alerts at system or application levels, see Alerts section for resolution steps.

In case the alerts still persist, perform the following:

- Collect the logs: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide*.
- Raise a service request: See My Oracle Support .

4.17.1 Handling N28 and N36 Interfaces Context Information during Subscription Failures

With this feature, PCF supports the N28 and N36 context information such as subscription information, policy and charging related information to be stored in PDS databases during lookup failures. In a scenario, where:

1. SMF initiates a session creation by sending SM create request with received DDN as IMS, the session ID is `session-id-ims` for a subscriber.
2. The fetch request for subscription details from the N28 and N36 fails. PDS does not store any context information for `session-id-ims`.
3. SMF initiates a session creation by sending SM create request with received DNN as internet, the session ID is `session-id-internet` for the same subscriber.
4. If the GET request for subscription details from the N28 and N36 is successful and the POST request for CHF or UDR subscription is also successful then PDS stores the information related to `session-id-internet` in the subscriber context information.
5. PCF receives a SM delete request for the `session-id-internet`.

Currently, PCF deletes the session `session-id-internet` information, even though the other session `session-id-ims` still exists for the subscriber.

This implementation ensures that PCF does not perform deletion of the message toward CHF or UDR prematurely, and keeps the information in the PDS until the subscriber goes away. This feature is implemented for PCRF Core and SM services.

To implement this feature, PCRF core services uses an existing `USER.ocsSpendingLimit.createContextOnFailure` flag and the following new flags:

- `USER.smPolicyData.createContextOnFailure`
- `USER.operatorSpecificData.createContextOnFailure`

If these flags are set to true, then the subscribers policy data and operator specific data are stored in a dummy context created by PDS service during subscription failures from CHF or UDR.

SM service uses an existing `USER.CREATE_CONTEXT_ON_FAILURE_OCS_DATA` flag and the following three new flags:

- `USER.CREATE_CONTEXT_ON_FAILURE_SM_POLICY_DATA`
- `USER.CREATE_CONTEXT_ON_FAILURE_CHF_DATA`
- `USER.CREATE_CONTEXT_ON_FAILURE_OPERATOR_SPECIFIC_DATA`

If these flags are set to true, then the subscribers policy data, operator specific data, and charging data are stored in a dummy context created by PDS service during subscription failures from CHF or UDR.

> **Note**
>
> This feature works only when subscription (subs-to-notify) flag for UDR data source is true. If the subscription flag is false, then PDS does not create any UDR related resources in the database.

For more information on CNC Console configurations, see "Advanced Settings" in PCF Session Management and PCRF Core Settings sections.

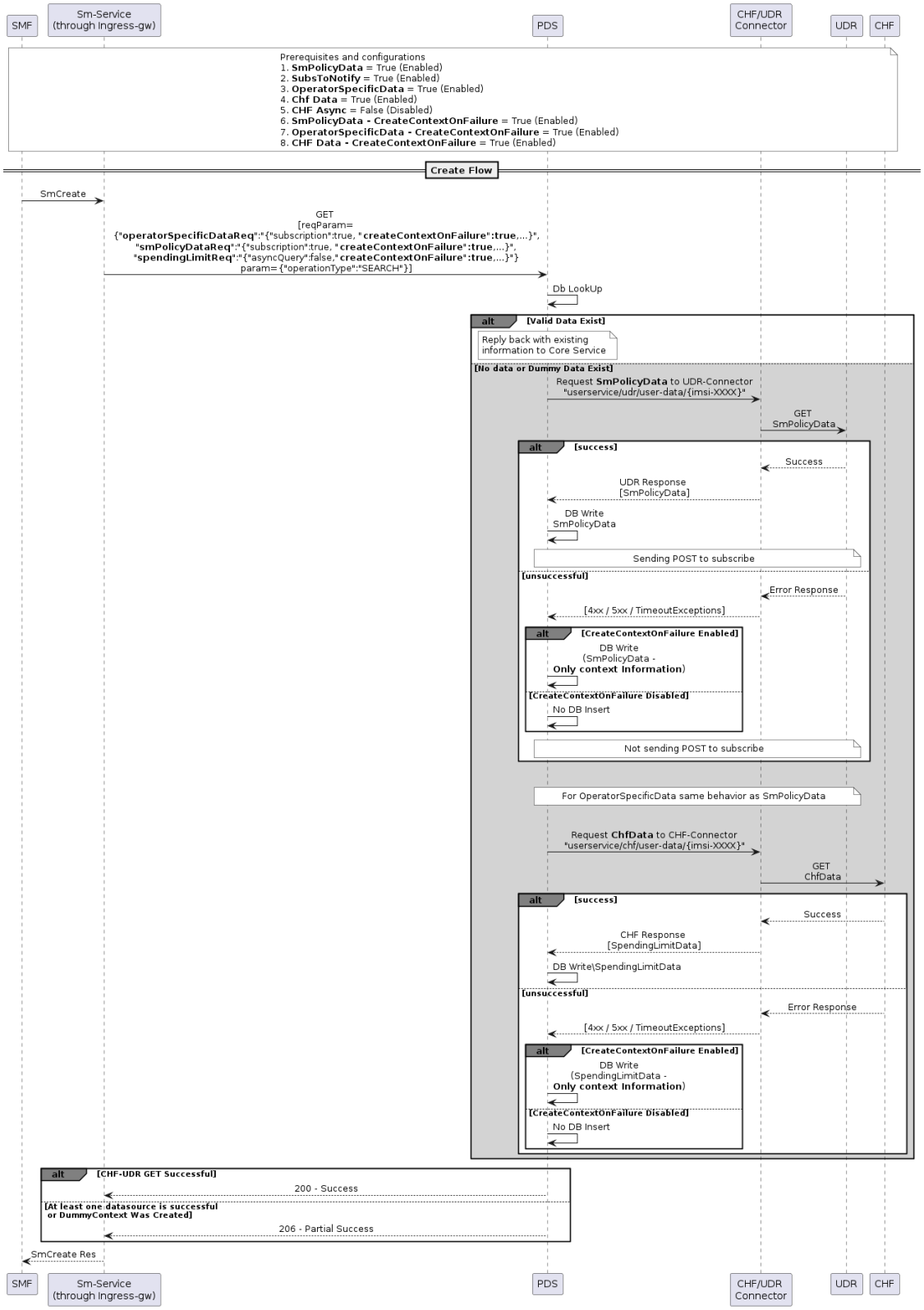
Call Flows

This section describes the procedure of dummy context creation by PDS service during subscription failure scenario in SM and PCRF Core services Create, Update, and Delete call flows.

PDS Dummy Context Creation in SM and PCRF Core Services Create Call Flow

This section describes the SM Create call flow and usage of the flag `createContextOnFailure` during CHF/UDR subscription failure. The same call flow applies to PCRF-Core (CCR-I) without SpendingLimitData (CHF).

Figure 4-11 PDS Dummy Context Creation during SM and PCRF Core Create Call Flow



1. The SMF sends a SM Create request to SM service via Ingress Gateway.

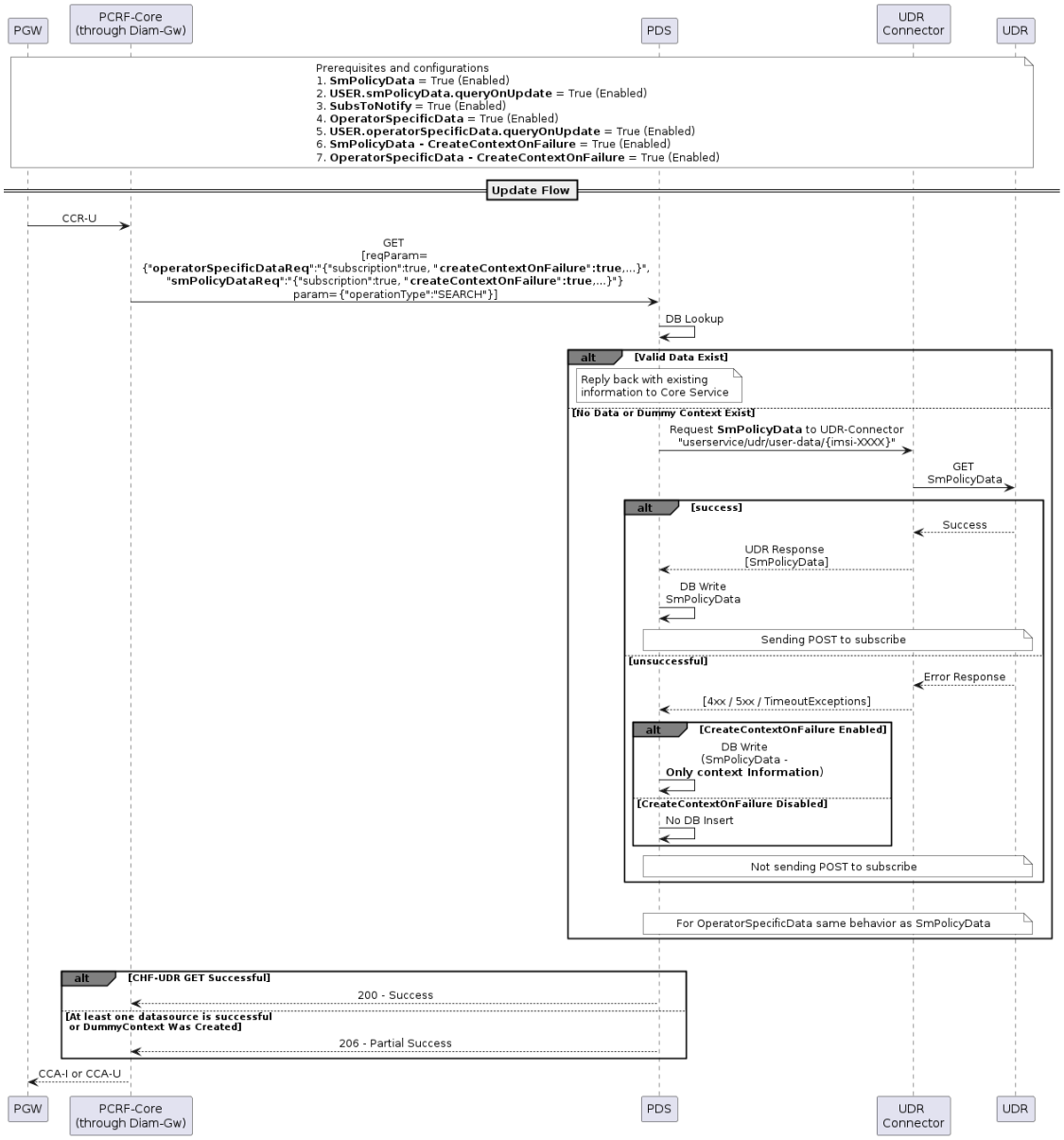
2. If UDR or CHF subscription and charging information is configured in PCF then SM service populates the SM session with these details and sends a request to PDS.
3. The PDS searches for the charging information in the database.
   - If valid data exists, then PDS responds to SM service with the data.
   - If dummy data exists or there is no data, then it sends a GET request for SmPolicyData/OperatorSpecificData toward UDR, and GET request for ChfData toward CHF.
     – UDR responds with SmPolicyData to PDS. This data is stored to the database by PDS.
     – PDS sends Subscription request to UDR. On receiving unsuccessful error response from UDR, PDS checks for the flag "CreateContextOnFailure". If it is set to true, then the SmPolicyData/OperatorSpecificData and ChfData are written to the dummy context by PDS. If the flag is false, there is no database operation done by PDS.
4. PDS sends successful `200 Success` response to SM service for CHF and UDR GET requests.
5. PDS sends partial successful `206 Partial Success` response to SM service on creation of either one datasource or dummy context.
6. SM service responds with SmCreate response to SMF.

> **Note**
>
> Upon dummy context creation for the data source, it will contain "lastErrorCode" parameter containing status code that provides error details. For example, on receveing 404 error Not Found from UDR, the "lasterrorcode" will be populated as - `lastErrorCode:"USER_NOT_FOUND" or "SUBSCRIPTION_NOT_FOUND"` depending on the type of operation performed.

PDS Dummy Context Creation in SM and PCRF Core Services Update Call Flow

This section describes the PCRF Core Update call flow and usage of the flag `createContextOnFailure` during CHF/UDR subscription failure. The same call flow applies to SM Update with SpendingLimitData (CHF).

Figure 4-12 PDS Dummy Context Creation during SM or PCRF Core Update Call Flow



1. The PGW sends a CCR-U, an Update request to PCRF core service via Diameter Gateway.
2. If UDR or CHF subscription is configured, then PCRF core service populates its session with these details and sends a request to PDS.
3. The PDS searches for the charging information in the database. If valid data exists, then PDS responds to PCRF core service with the data.
4. The PDS searchs for the charging information in the database and if no data or dummy data exists, then
   - If valid data exists, then PDS responds to PCRF core service with the data.
   - If dummy data exists or there is no data, then it sends a GET request for SmPolicyData/OperatorSpecificData toward UDR.

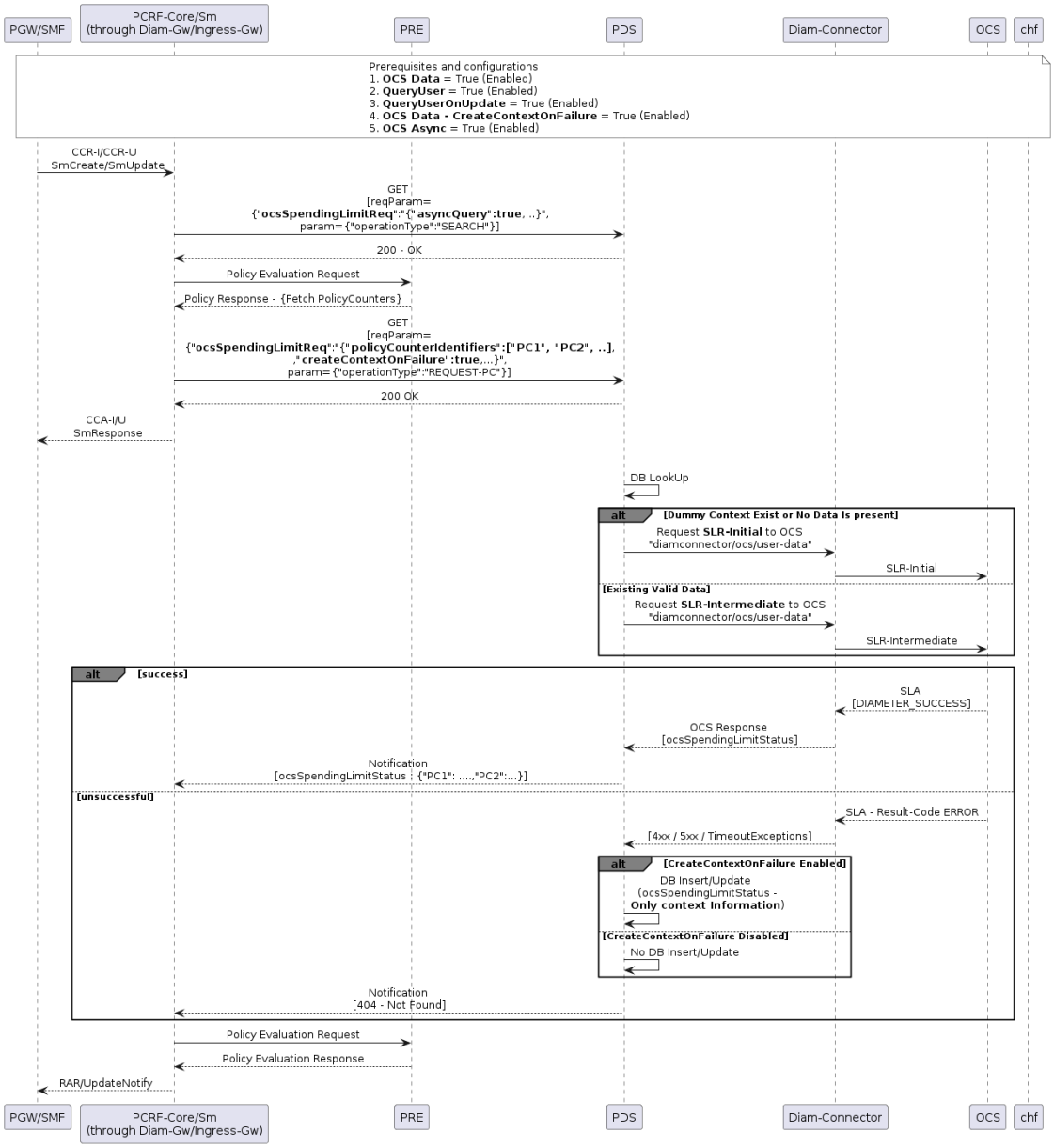
- UDR responds with SmPolicyData to PDS. This data is stored to the database by PDS.
- PDS sends Subscription request to UDR. On unsuccessful error response from UDR, PDS checks for the flag "CreateContextOnFailure" value. If it is true, then the SmPolicyData/operatorSpecificData is written to dummy context by PDS. If the flag is false, there is no database operation done by PDS.

5. PDS sends successful `200 Success` response to PCRF Core service for UDR GET request.
6. PDS sends partial successful `206 Partial Success` response to SM service on creation of either one datasource or dummy context.
7. PCRF Core Service responds with CCA-I or CCA-U response to PGW.

PDS Dummy Context Creation in SM Create/Update Call Flow with OCS in Asynchronous Mode

This section describes PCRF Core CCR-I/CCR-U call flow and usage of the flag `createContextOnFailure` during CHF/UDR subscription failure. For this call flow, the Online Charging System (OCS) is asynchronously enabled using the flag `OCS Aync`. The same call flow applies to SM Update with SpendingLimitData (CHF).

Figure 4-13 PDS Dummy Context Creation during SM and PCRF Core Create/Update Call Flow with OCS in Async Mode



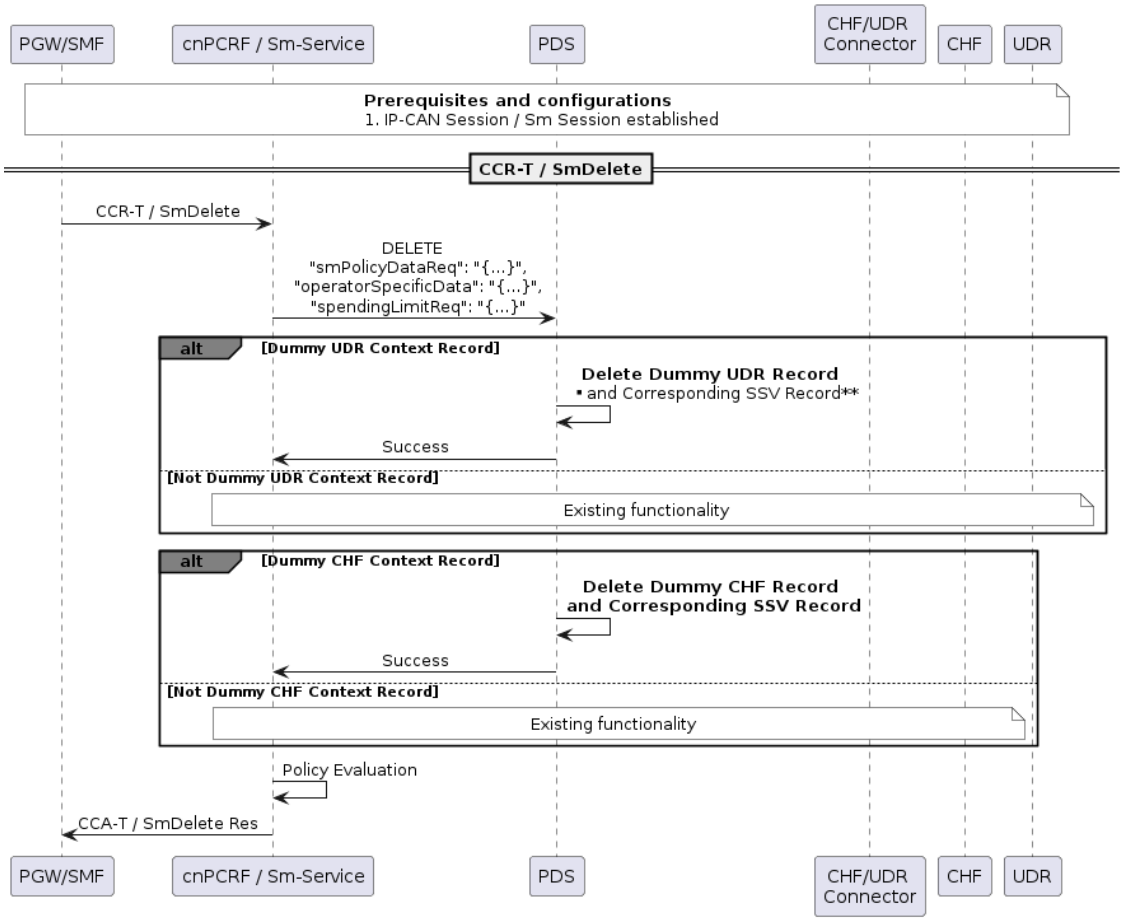
1. The PGW/SMF sends a CCR-I/SM Create or CCR-U/SM Update requests to PCRF core/SM services via Diameter Gateway/Ingress Gateway.
2. The PCRF core/SM service sends a GET request with `asynchQuery` enabled in OCS spending limit header parameter toward PDS.
3. PDS sends successful `200 Success` response to PCRF core/SM service.
4. The PCRF core/SM service sends a Policy evaluation to PRE.
5. The PRE service responds with fetch policy counters request to PCRF core/SM service.
6. The PCRF core/SM service sends a GET request with `createContextOnFailure` enabled in spending limit header parameter toward PDS.

7. PDS sends successful `200 Success` response to PCRF core/SM service.
8. The PCRF core/SM service sends a CCA-I/SMCreateResponse or CCA-U/SMUpdateResponse toward PGW/SMF.
9. The PDS performs Database lockup:
   - If dummy context exists, and there is no valid data, then PDS sends SLR-Initial request toward OCS.
   - If valid data exits, then PDS sends SLR-Intermediate request toward OCS.
10. OCS sends successful "ocsSpendingLimitStatus" response to PDS. PDS sends a notification about "ocsSpendingLimitStatus" to PCRF Core/SM service.
11. OCS sends unsuccessful error response to PDS. If `createContextOnFailure` is enabled then ocsSpendingLimitStatus context information is written to database by PDS. If the flag is false, there is no database operation done by PDS.
12. PDS sends a notification `404 Not Found` request to PCRF Core/SM service.
13. PCRF Core/SM services sends policy evaluation request to PRE service.
14. PRE service sends policy evaluation response to PCRF Core/SM service.
15. PCRF Core/SM service sends RAR/UpdateNotify response to PGW/SMF.

PDS Dummy Context during SM Delete or PCRF Core CCR-T Call Flow

This section describes the SM Delete/CCR-T call flow and usage of the flag `createContextOnFailure` during CHF/UDR subscription failure.

Figure 4-14 PDS Dummy Context during SM Delete or PCRF Core CCR-T Call Flow



1. The PGW/SMF sends a CCR-T/SM Delete request to PCRF Core/SM services via Diameter Gateway/Ingress Gateway.
2. PCRF core/SM services send delete request to PDS.
3. For the given subscriber, if the dummy context does not exists, then PDS deletes it and also the corresponding Subscriber State Variable (SSV).
4. If dummy context for related to the specific subscriber in the request exists then PDS deletes it and the corresponding Subscriber State Variable (SSV).
5. PDS sends successful `200 Success` response to PCRF core/SM services.
6. PCRF core/SM services perform policy evaluation and send the result as CCA-T/SM Delete response to PGW/SMF.

4.18 Enhancement to PRE Metrics

The following Policy blocks have been added to PRE to evaluate policy block execution counter and policy blocks execution time.

- Increment Counter Label
- Time

For more information, see "Public Category" section in Oracle Communications Cloud Native Core, Converged Policy Design Guide.

To enable the PRE metrics, enable the **Enable Metrics** switch in the Policy Engine page under Service Configurations. By default, this switch remains disabled. For more information, see Policy Engine.

The following metrics have been added in PRE service for this feature:

- occnp_block_counter_label
- occnp_block_exec_time_ns

For more information, see PRE Metrics.

4.19 Enhancement to PCF Resiliency

PCF is enhanced to provide the mechanism in consumer service pods to cache the last successful producer pod information and use it when the subsequent producer pod discovery results in failure. Hence, the inter service communication continues to run even when the communication between the Kubernetes services and pods is down.

4.20 Support for cnDBTier Functionalities in CNC Console

With the implementation of this feature, cnDBTier functionalities are integrated into the CNC Console, and Policy users can view specific cnDBTier functions, such as checking the cnDBTier version, status of cnDBTier clusters, and georeplication status on the CNC Console.

> **Note**
>
> This **cnDBTier** options can be accessed only through CNC Console.

The following cnDBTier functionalities are read only and can be viewed on the CNC Console:

- Backup List: It displays the details of stored backups, such as the ID and size of the backup.
- cnDBTier version: It displays the cnDBTier version.
- Database Statistics Report: It displays the number of available database.
- Georeplication Status:
  - Real Time Overall Replication Status: It displays the overall replication status in multisite deployments. For example, in a four-site deployment, it provides the replication status between the following sites: site1-site2, site1-site3, site1-site4, site2-site3, site2-site4, and site2-site1. This is applicable for all other sites.
  - Site Specific Real Time Replication Status: Itdisplays the site-specific replication status.
- HeartBeat Status: It displays the connectivity status between the local site and the remote site to which Policy is connected.
- Local Cluster Status: It displays the status of the local cluster.
- On-Demand Backup: It provides options to initiate as well as the display the status of the on-demand backup. It also displays the status of initiated on-demand backups.

Managing cnDBTier Functionalities at CNC Console

Enable
This feature is enabled automatically when cnDBTier is configured as an instance during the CNC Console deployment. For more information about integrating cnDBTier functionalities in CNC Console, see *Oracle Communications Cloud Native Core, cnDBTier User Guide*.

Configure

You can view cnDBTier functionalities at CNC Console in the Viewing cnDBTier Functionalities in CNC Console section.

Maintain

If you encounter alerts at the system level, see the Alerts section for resolution steps.

In case the alerts persist, perform the following tasks:

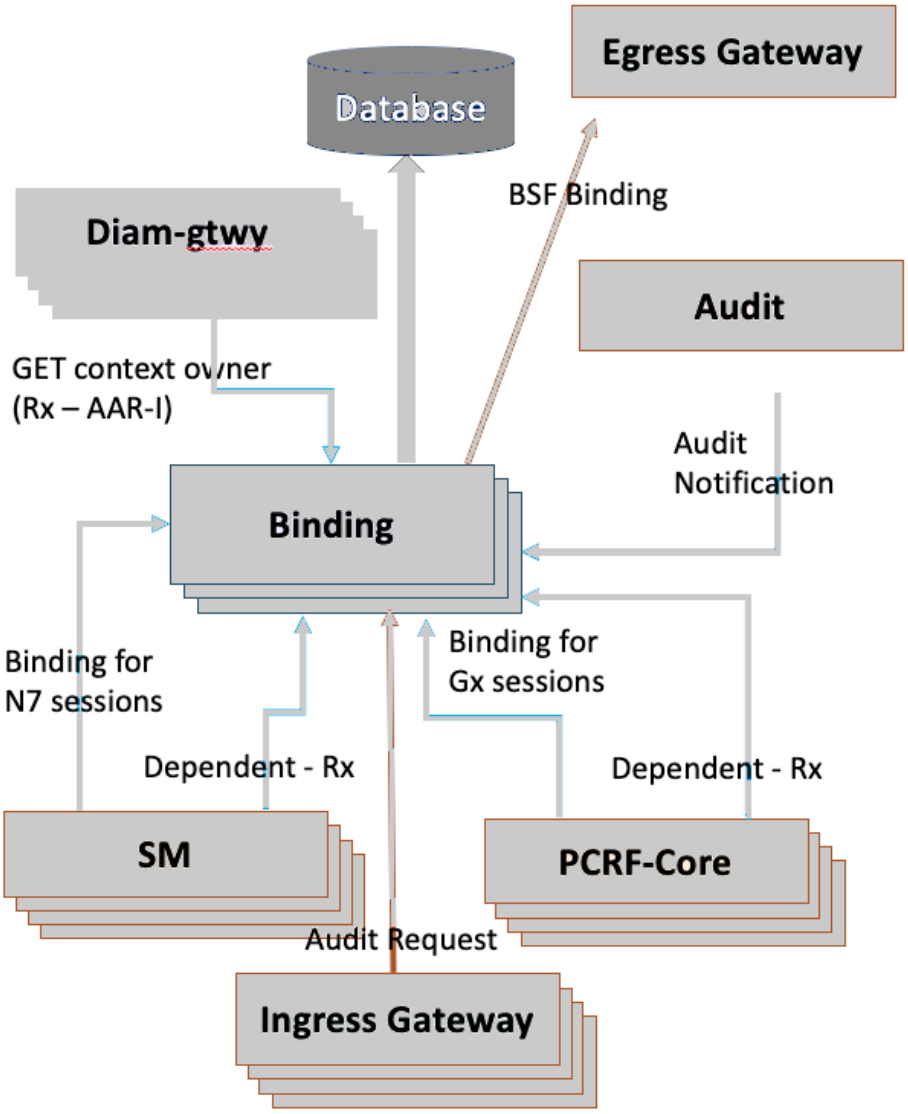
1. **Collect the logs**: For information about how to collect logs, see *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide*.
2. **Raise a service request**: For information about how to raise a service request, see My Oracle Support.

4.21 Binding Service Pod Congestion Control

Binding Service, a Policy microservice provides:

- session binding information on receiving binding request from SM service.
- subscriber/Data Network Name (DNN) session limit check across Gx and N7 session or within multiple sessions for a 4G/5G deployments.
- context owner information to Diameter Gateway for routing Rx messages for a 4G/5G deployments.

Figure 4-15 Binding Service and Other Policy Services



The above diagram shows Binding service interaction with Policy SM, PCRF Core, cnDBTier, Audit, Diameter Gateway and Egress Gateway services. At times, an excessive traffic from these services can be observed in the network, which can result in a high CPU utilization, high memory utilization. This can cause performance degradation in Binding service responses and eventually reach a state of service unavailability. Congestion control is used in order to help in identifying such conditions and invoke rules that address these situations when these load conditions persist.

Pod Congestion Control Mechanism

The Pod congestion control mechanism involves:

1. Determining Pod Congestion State
2. Triggering Pod Congestion Control

Determining Pod Congestion State

Binding service pods exist in any of the following five states at any given time:

- Normal
- Danger of Congestion (DOC)
- CONGESTION_L1
- CONGESTION_L2
- CONGESTED

Figure 4-16 Different Pod Congestion States

Periodically, the state of the pod's congestion gets determined. This interval is configurable, and the default setting is 200 milliseconds.

The pod's state gets determined by considering the following points.

1. Calculate the congestion state for the following resources:
   a. **Queue**: For the DOC, CONGESTION_L1, CONGESTION_L2 and CONGESTED pod states, compare the number of pending messages in the queue with the configured pending messages threshold.
   b. **CPU**: The CPU usage for congestion state is calculated by comparing the CPU usage of the Container (monitored using cgroup parameter - `cpuacct.usage` that provides current cpu usage in nanoseconds) with the configured threshold.
2. The congestion state for pod gets assigned a maximum congested state based on the congestion state of the resources.

The Binding service pod's can be in following Congestion states:

Table 4-10 Binding service Congestion States

Congestion States	CPU Count	Queue Count
DANGER_OF_CONGESTION (DOC)	70	120
CONGESTION_L1	75	150
CONGESTION_L2	80	180
CONGESTED	85	200

Triggering Pod Congestion Control

Every time Binding service receives requests from other services, it checks for the current congestion state of the pod. The Congestion Control mechanism is triggered if the pod's congestion state is in DOC or Congested_L1 or Congested_L2 or Congested.

The requests to the Binding service might have priority included as `oc-message-priority` attribute in the request header. The priority value ranges between 0 to 100 with 0 being the highest and 100 being the lowest priority.

> **Note**
>
> Currently, the downstream services do not propagate the `oc-message-priority` header to Binding service and will be implemented in future releases.

Priority-Based Load Shedding

Based on the pods current congestion state a load shedding rule is applied to perform priority-based load shedding. The load shedding rule is based on message priority, e.g., when the Binding service Pod state is CONGESTED, then discard messages with priority 30. It determines if the message with the assigned priority should be rejected or accepted.

These rules get configured per congestion state. If there are no rules configured for a congestion state, then Binding service accepts the request as a default behavior. The user can customize the result codes for the rejected requests when configuring the load rules. The default result code is `503 Service Unavailable`.

The default load shedding rules for Binding service:

```
- state: DANGER_OF_CONGESTION
      discardPriority: 30
- state: CONGESTION_L1
      discardPriority: 27
- state: CONGESTION_L2
      discardPriority: 24
- state: CONGESTED
      discardPriority: 20
```

When Binding service is in congestion state, it's response can be configured using Binding service Advanced settings in CNC Console, using the key CONGESTION_RESPONSE_CODE. This key is used to configure for response code of the messages that is rejected by the Binding service due to pod congestion state. By default Binding Service responds with a response code of 503. The response code configured should be 5xx error status only. Following are the list of configurable keys that can be added to set the message priority:

Table 4-11 Configuring Message Priority

Key	Default Value	Allowed Values
AUDIT_MESSAGE_PRIORITY	30	0-100
BSF_AUDIT_MESSAGE_PRIORITY	27	0-100
DEPENDENT_CONTEXT_BINDING_REGISTER_MESSAGE_PRIORITY	24	0-100
DEPENDENT_CONTEXT_BINDING_DEREGISTER_MESSAGE_PRIORITY	16	0-100
DEPENDENT_CONTEXT_BINDING_FIND_CONTEXT_OWNER_MESSAGE_PRIORITY	16	0-100
SESSION_BINDING_REGISTER_MESSAGE_PRIORITY	24	0-100

Table 4-11 (Cont.) Configuring Message Priority

Key	Default Value	Allowed Values
SESSION_BINDING_UPDATE_MESSAGE_PRIORITY	20	0-100
SESSION_BINDING_DEREGISTER_MESSAGE_PRIORITY	16	0-100
SESSION_BINDING_SEARCH_MESSAGE_PRIORITY	30	0-100
SESSION_BINDING_FIND_CONTEXT_OWNER_MESSAGE_PRIORITY	16	0-100
SESSION_BINDING_CLEANUP_MESSAGE_PRIORITY	30	0-100
BSF_SESSION_UPDATE_MESSAGE_PRIORITY	20	0-100
STALE_SESSION_TRACKER_REFRESH_MESSAGE_PRIORITY	30	0-100

Note

- When upgrading from 24.1.x to 24.2.x or higher version then user has to update custom Threshold profiles manually, by taking new default values as a reference. When upgrading from versions lower than 24.1.0 to 24.2.x or higher version then the DEFAULT profile will be activated with new values.
- When exporting data from 24.1.x and importing it in 24.2.x or higher version then user has to update the custom Threshold profiles value manually by taking new default values as a reference before import.

Congestion Control Call Flow/Scenario

For the below call flows, consider the following Load shedding rule configuration:

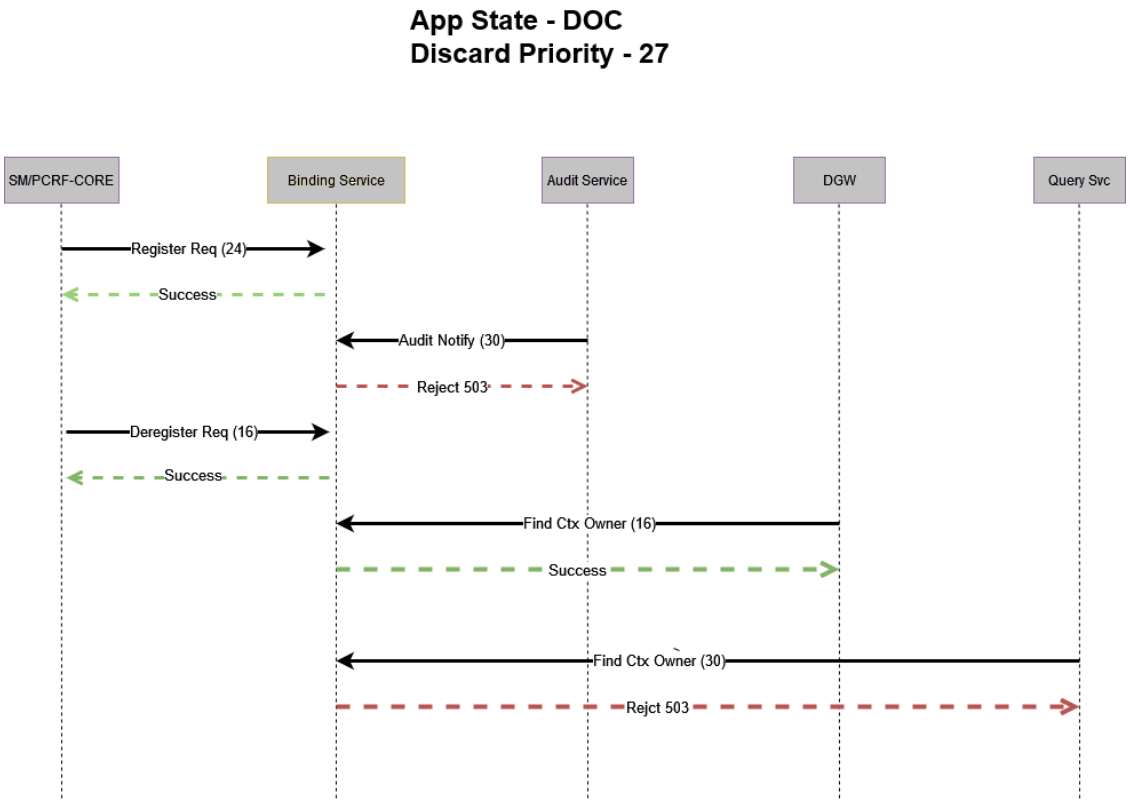
Table 4-12 Load Shedding Rule

Congestion State	Discard Priority
DOC	27
CONGESTION_L1	23
CONGESTION_L2	18
CONGESTED	10

Binding Service Call Flow in Doc State

The call flow describes the Binding service congestion control mechanism handling the requests from the other Policy services such as SM/PCRF core, Audit, Diameter Gateway, and Query services when it is in Danger of Congestion (DOC) state.

Figure 4-17 Binding Service Pod in DOC State

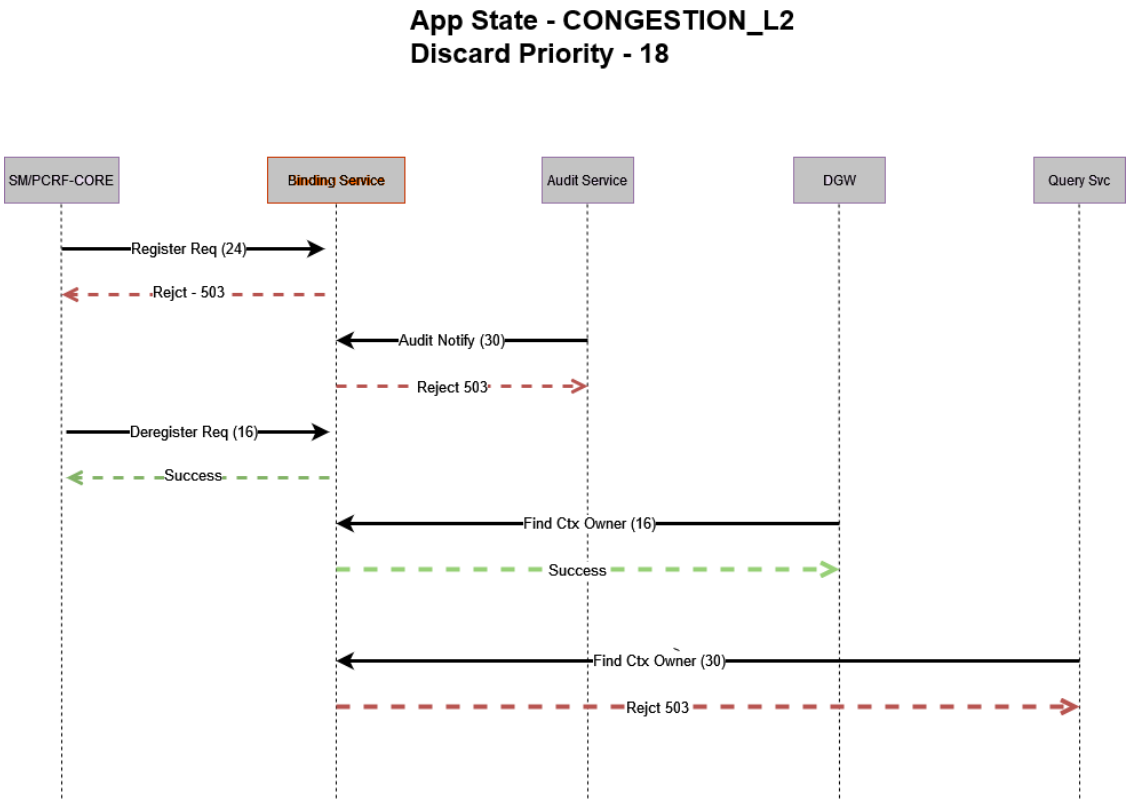


- SM/PCRF-Core services sends a register request with discard priority set to 24 to Binding service and it responds back with success status.
- Audit service sends a audit notification request with discard priority set to 30 to Binding service and it responds back with a reject 503 status.
- SM service sends a de-register request with discard priority set to 16 to Binding service and it responds back with success status.
- Diameter Gateway service sends a find context owner request with discard priority set to 16 to Binding service and it responds with a success status.
- Query Gateway service sends a find context owner request with discard priority set to 30 to Binding service and it responds back with a reject 503 status.

Binding Service Call Flow in Congestion_L2 State

The call flow describes the Binding service congestion control mechanism handling the requests from the other Policy services such as SM/PCRF core, Audit, Diameter Gateway, and Query services when it is in Congestion_L2 state.

Figure 4-18 Binding Service in Congestion_L2 State



- SM/PCRF-Core service sends a register request with discard priority set to 24 to Binding service and it responds back with a reject 503 status.
- Audit service sends a audit notification request with discard priority set to 30 to Binding service and it responds back the a reject 503 status.
- SM service sends a de-register request with discard priority set to 16 to Binding service and it responds back with success status.
- Diameter Gateway service sends a find context owner request with discard priority set to 16 to Binding service and it responds back with success status.
- Query Gateway service sends a find context owner request with discard priority set to 30 to Binding service and it responds back with a reject 503 status.

Managing Binding service Pod Congestion Control

Enable

By default, the Pod Congestion control is disabled for Binding service. You can enable this feature using CNC Console or REST API for Policy.

Configure Using CNC Console

To enable the feature using CNC Console set the **Enable** parameter in **Settings** page under **Congestion Control** for **Overload and Congestion Control Configurations**.

Perform the Binding service Congestion Control feature configurations on the Settings, Threshold and Load Shedding Rules in CNC Console as described in Congestion Control section.

Perform the Binding service Congestion Control feature discard message configurations at Advanced Settings in CNC Console as described in Binding Service section.

Configure Using REST API

Perform the feature configurations as described in "Congestion Control" section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Observability

Metrics:

Following metrics were updated in the Pod Congestion Metrics section.

- `occnp_pod_congestion_state`
- `occnp_pod_resource_stress`
- `occnp_pod_resource_congestion_state`
- `pod_cong_state_report_total`
- `pod_resource_congestion_state_report_total`
- `http_congestion_message_reject_total`

Alerts

The following alerts generated for this feature:

- POD_CONGESTED
- POD_CONGESTION_L2
- POD_CONGESTION_L1
- POD_DANGER_OF_CONGESTION
- POD_PENDING_REQUEST_CONGESTED
- POD_PENDING_REQUEST_CONGESTION_L2
- POD_PENDING_REQUEST_CONGESTION_L1
- POD_PENDING_REQUEST_DANGER_OF_CONGESTION
- POD_CPU_CONGESTED
- POD_CPU_CONGESTION_L2
- POD_CPU_CONGESTION_L1
- POD_CPU_DANGER_OF_CONGESTION

For more information about alerts, see Common Alerts.

Maintain

Error logs are generated when the system is congested and the actions are taken to bring the system back to normal. Warning logs are generated to indicate the congestion level. However, error logs are not generated when messages are rejected to avoid additional resource usage to write error logs.

If you encounter alerts at system or application levels, see Alerts section for resolution steps.

In case the alerts still persist, perform the following:

- Collect the logs: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide*.

- Raise a service request: See My Oracle Support.

4.22 Support for Non-SUPI based On-Demand Discovery Caching of NF Profiles

Policy supports the caching of NF profiles at NRF Client received from non-SUPI based on-demand discovery from NRF. Caching the NF profiles at NRF Client avoids discovering the NF profiles from NRF for every new call flow.

- N15 - AMF Notification for AM Policy (NFSetid based discovery for retry on Notification Request failure)
- N15 - AMF Notification for UE Policy (NFSetid based discovery for retry on Notification Request failure)
- N7 - SMF Notification for SM Policy (NFSetid based discovery for retry on Notification Request failure)
- N15 - AMF discovery, AMF as a producer for UE Policy N1N2 Message Subscribe (GUAMI, amf-region-id,amf-set-id)
- N15 - AMF rediscovery, AMF as a producer for UE Policy N1N2 Message Transfer failure retry (NFSetid based discovery)
- UDR Rediscovery, UDR as a producer for SM, AM and UE Policy session, when POST(if considered subsequent), PUT, PATCH, Delete request failure, (NFSetid based discovery)

This allows NRF client to map and cache discovery query parameters against their on-demand discovery responses, allowing NRF Client to reuse those responses until they expire and reduces the number of requests sent to NRF for such discoveries.

Policy supports:

- caching the discovered producer/Notify Consumers profiles from NRF along with its query parameter for on-demand discovered NFs.
- rediscovering the NF Profiles whose TTLs have expired, when there is an on-demand discovery request for the same. That is, rediscovering the NF profiles based on the query parameter, when required during a signaling or call flow, if the validity period of the cached discovery response has expired. The expired records are updated with the recently rediscovered records after rediscovery is successful with 2xx response code.
- using the expired discovered profiles, which are in Registered state, if PCF cannot reach any NRFs during rediscovery.

The query from SM service, AM service, UE Policy service towards NRF Client can include the following two headers to support this feature:

- OC-Force-Rediscovery: Indicates whether to cache the NF profiles received from NRF in NRF Client or to skip caching and receive the response directly from NRF.

  Value of this parameter can be 0 or 1. By default, value of this parameter is set to 0.

  - When the value of OC-Force-Rediscovery parameter is set to 0, it enables the feature and caches the NF profiles details received from NRF.
  - When the value of OC-Force-Rediscovery parameter is set to 1, it disables the feature and retrieves the NF profiles directly from NRF while responding to the query from the backend services.
- OC-Retention-Period: Indicates the time a record is allowed to stay in database until the expiry period. This parameter accepts an integer value to indicate the retention period in

milliseconds. When retention period is not configured/saved as null, backend will not send this header to NRF connector. By default, value of this parameter is set to 0.

Important

If occnp_nrf_client database is not yet created on NRF Client, you must manually create it. This database is used to store the cached profiles on NRF Client.

For details on how to create the database, see *Configuring Database, Creating Users, and Granting Permissions* section in *Oracle Communications Cloud Native Core, Installation, Upgrade, and Fault Recovery Guide* .

These headers can be configured by applying the configurations under NF Discovery Settings group in NF Communication Profiles for Common Data in Service Configuration.

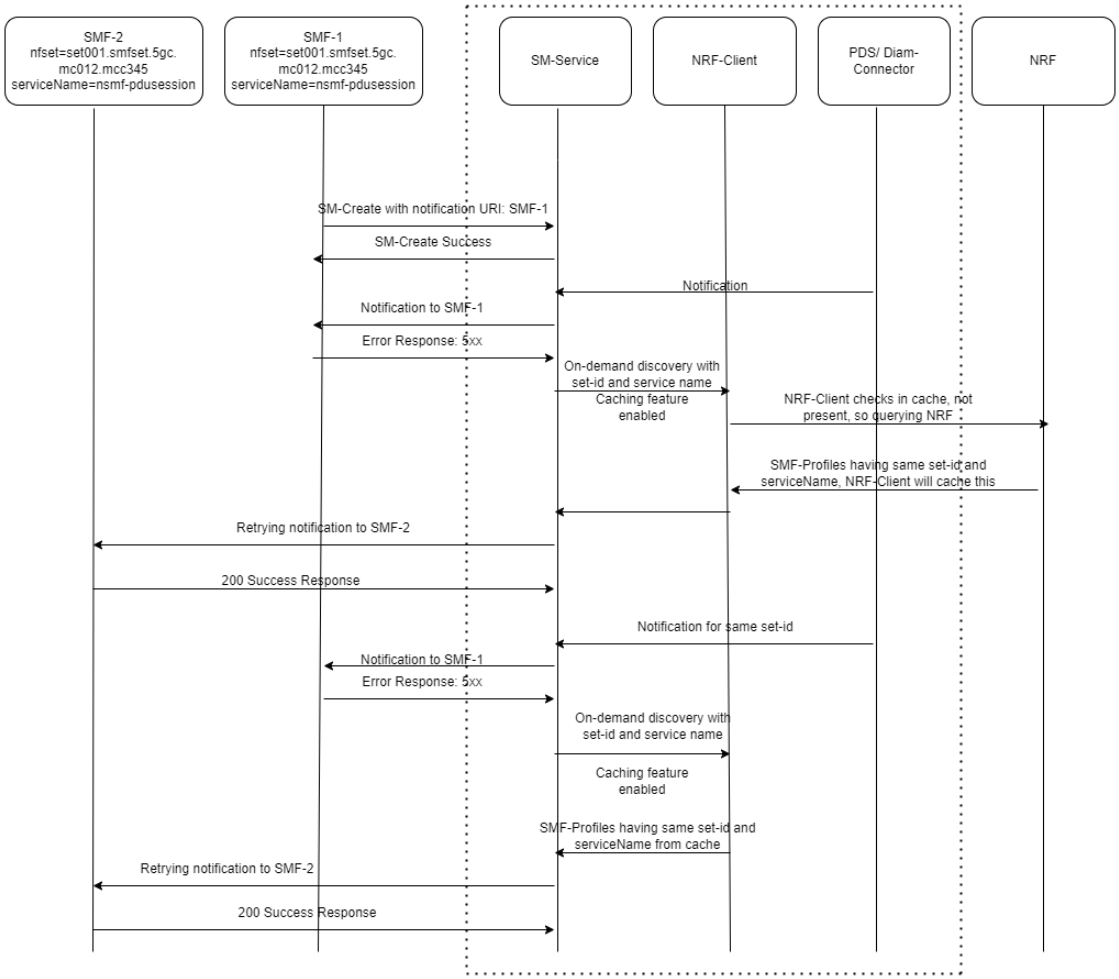
The configurations are saved in common.public.communication-profile topic in the config server.

Note

The NF communication profile must be created and oc-forced-rediscovery, retentionperiod must be configured before attaching the NF communication profile to service configuration page of SM service, AM service, UE Policy service, and User service.

Call Flow

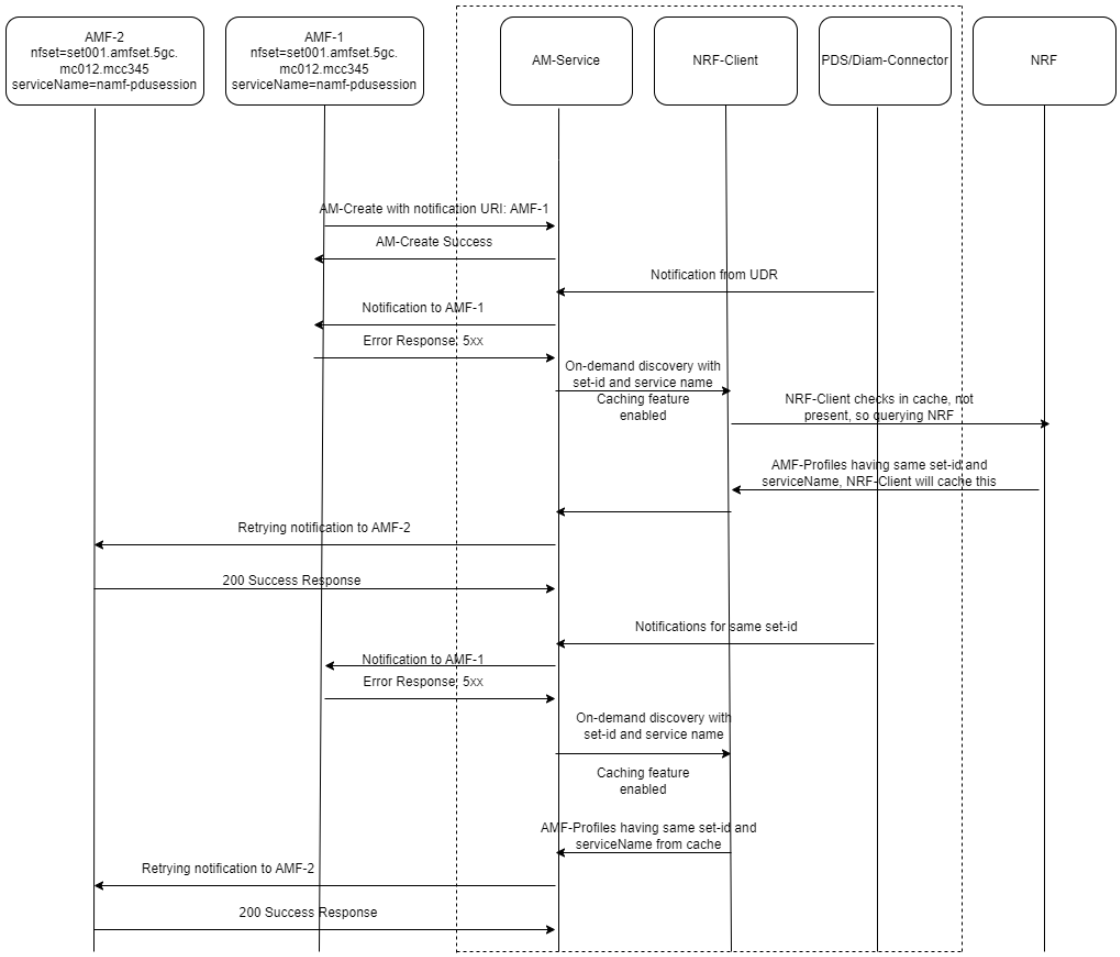
Figure 4-19 SMF Notification for SM Policy (NFSetid based discovery for retry on Notification Request failure)



1. SMF1 sends an SM Create request to SM service.
2. After the successful creation, SM service responds to SMF1 with a successful message.
3. Whenever there is a notification from UDR regarding any change in the user profile, PDS forwards the notification to SM service.
4. SM service forwards the notification to SMF1 and receives a 5xx error.
5. If the caching feature is enabled, SM service initiates an on-demand discovery caching with the given set ID and Service name and sends the request to NRF client.
6. As this is the first request and the data is not yet cached, NRF client sends the request to NRF to fetch the SMF profile information.
7. NRF responds with the SMF profile information for the given setID and service name.
8. NRF client caches these details in the database and sends the details to SM service.
9. SM service forwards the details to SMF2, which in turn responds with a 200 ok successful message.

10. Whenever PDS forwards an update notification from UDR to SM service.
11. SM service forwards the notification to SMF1 and receives an error in response.
12. SM service sends an on-demand discovery request to NRF client.
13. Now that the data is already cached for the given setID and service name, NRF client fetches the SMF Profiles from the cache and responds to SM service.
14. SM service sends the notification to SMF2 and receives a 200 ok successful message.

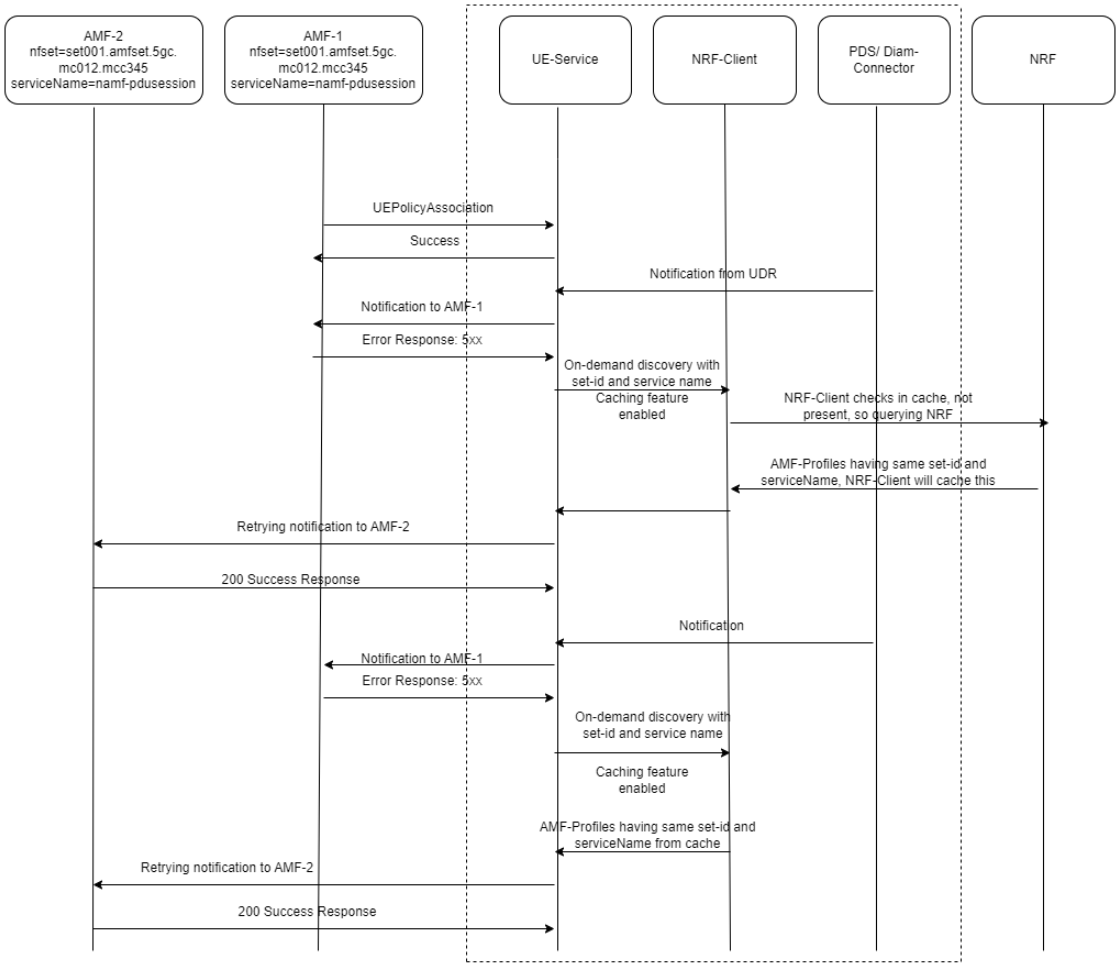
Figure 4-20 AMF Notification for AM Policy (NFSetid based discovery for retry on Notification Request failure)



1. AMF1 sends an AM Create request to AM service.
2. After the successful creation, AM service responds to AMF1 with a successful message.
3. Whenever there is a notification from UDR regarding any change in the user profile, PDS forwards the notification to AM service.
4. AM service forwards the notification to AMF1 and receives a 5xx error.
5. If the caching feature is enabled, AM service initiates an on-demand discovery caching with the given set ID and Service name and sends the request to NRF client.
6. As this is the first request and the data is not yet cached, NRF client sends the request to NRF to fetch the AMF profile information for the give set ID and Service name.

7. NRF responds with the AMF profile information for the given setID and service name.
8. NRF client caches these details in the database and sends the details to AM service.
9. AM service forwards the details to AMF2, which in turn responds with a 200 ok successful message.
10. Whenever PDS forwards an update notification from UDR to AM service.
11. AM service forwards the notification to AMF1 and receives an error in response.
12. AM service sends an on-demand discovery request to NRF client.
13. Now that the data is already cached for the given setID and service name, NRF client fetches the AMF profile information from the cache and responds to AM service.
14. AM service sends the notification to AMF2 and receives a 200 ok successful message.

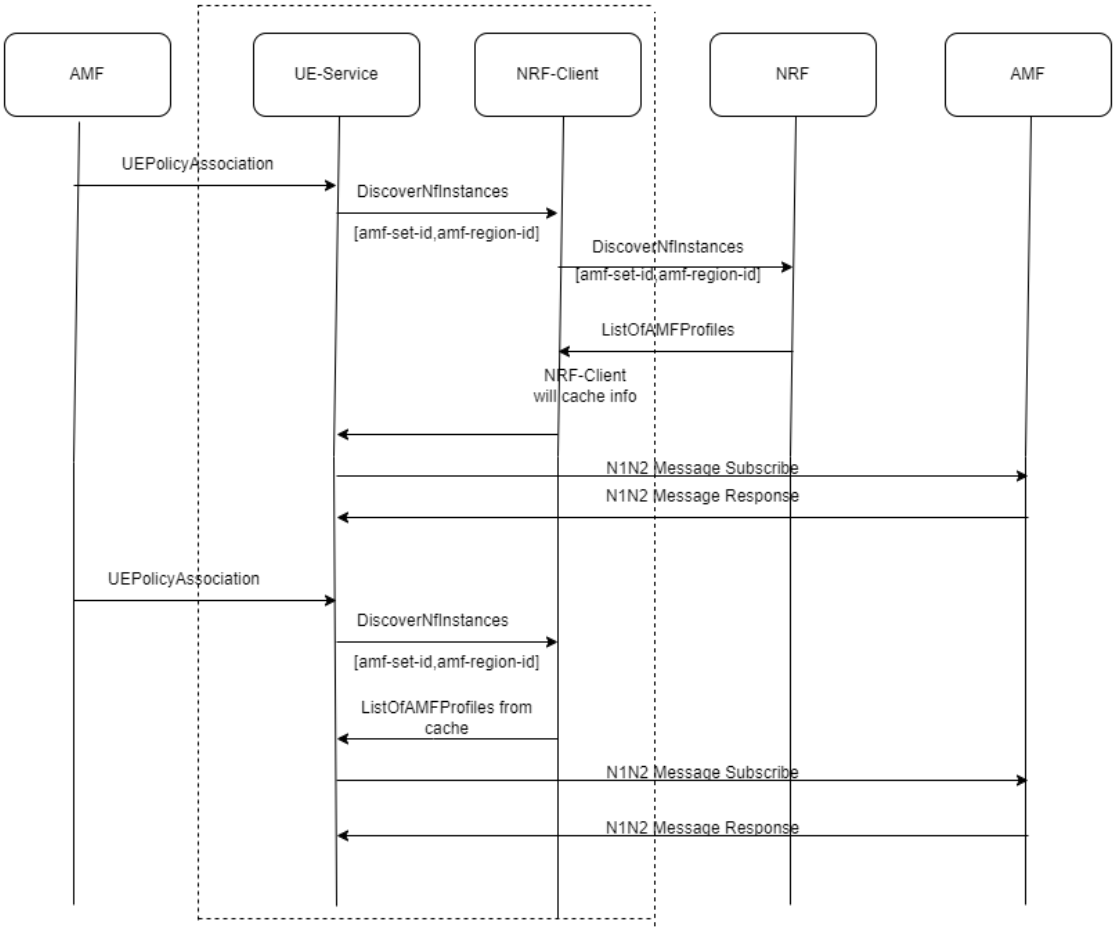
Figure 4-21 AMF Notification for UE Policy (NFSetid based discovery for retry on Notification Request failure)



1. AMF1 sends an UE Create request to UE Policy service.
2. After the successful creation, UE Policy service responds to AMF1 with a successful message.
3. Whenever there is a notification from UDR regarding any change in the user profile, PDS forwards the notification to UE Policy service.

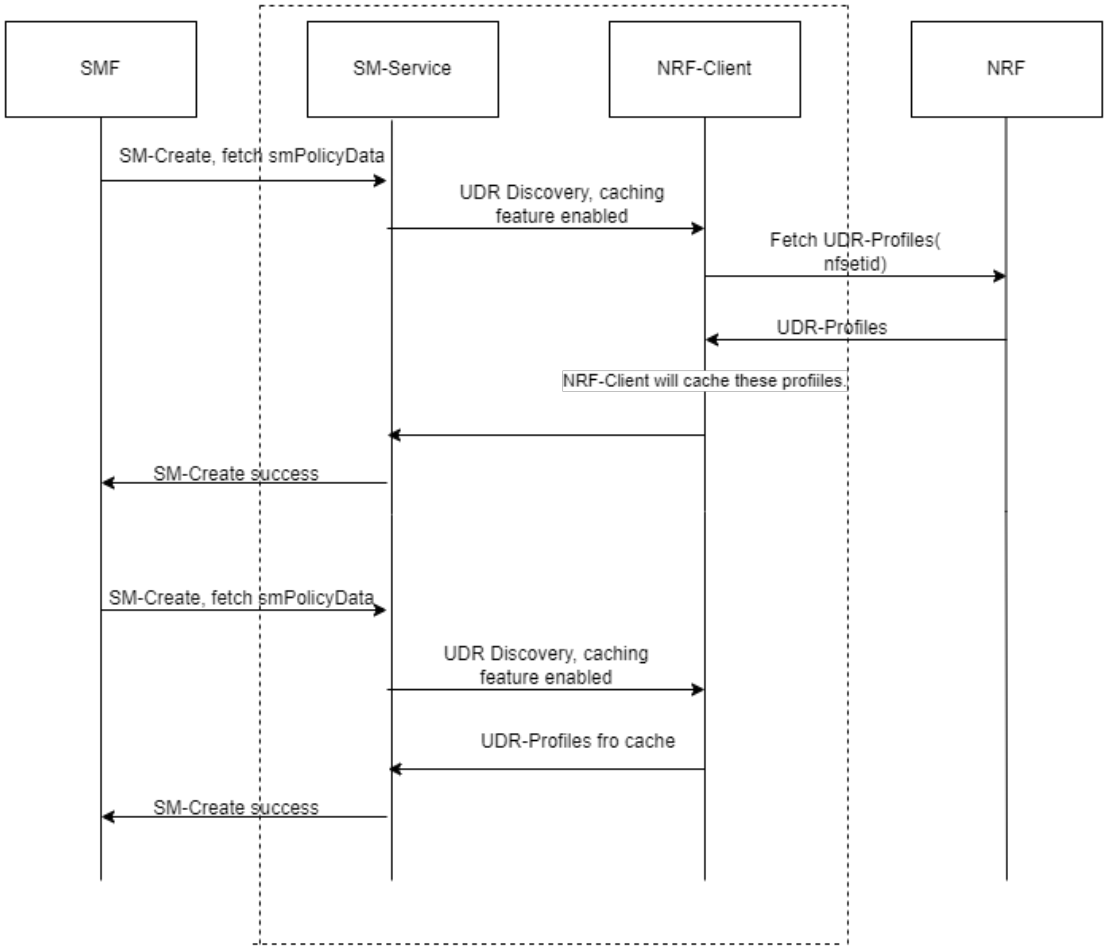
4. UE Policy service forwards the notification to AMF1 and receives a 5xx error.
5. If the caching feature is enabled, UE Policy service initiates an on-demand discovery caching with the given set ID and Service name and sends the request to NRF client.
6. As this is the first request and the data is not yet cached, NRF client sends the request to NRF to fetch the AMF profile information for the give set ID and Service name.
7. NRF responds with the AMF profile information for the given setID and service name.
8. NRF client caches these details in the database and sends the details to UE Policy service.
9. UE Policy service forwards the details to AMF2, which in turn responds with a 200 ok successful message.
10. Whenever PDS forwards an update notification from UDR to UE Policy service.
11. UE Policy service forwards the notification to AMF1 and receives an error in response.
12. UE Policy service sends an on-demand discovery request to NRF client.
13. Now that the data is already cached for the given setID and service name, NRF client fetches the AMF profile information from the cache and responds to UE Policy service.
14. UE Policy service sends the notification to AMF2 and receives a 200 ok successful message.

Figure 4-22 AMF Discovery/ Re-Discovery



1. AMF Sends a UEPolicyAssociation request to UE Policy service.
2. UE Policy service sends a request to NRF Client to discover the NF instances using amf-set-id and amf-region-id.
3. NRF client forwards the request to NRF.
4. NRF responds to NRF client with the list of AMF Profiles matching the given amf-set-id and amf-region-id.
5. NRF Client caches these AMF Profiles and then responds to UE Policy service.
6. UE Policy service sends a N1N2 message subscription request to the AMF matching the amf-set-id and amf-region-id and receives a successful response.
7. When UE Policy service receives the next UEPolicyAssociation create request for the same amf-set-id and amf-region-id, it sends a request to NRF client to fetch the AMF profiles.
8. As the data is already cached, NRF client fetches the AMF profiles from its cache and responds to UE Policy service.
9. UE Policy service sends the N1N2 message subscription request to the required AMF and receives a successful response.

Figure 4-23 Non-SUPI based on-demand discovery caching for User Service



1. SMF sends an SM Create request to SM service.
2. SM service sends a request to NRF client to fetch the UDR profiles details.
3. NRF client sends the request to NRF and receives the list of UDR profiles matching the given nfsetID.
4. If the caching feature is enabled, NRF client caches these UDR profiles and then forwards the details to SM service.
5. After the successful creation of SMPolicyAssociation, SM service responds to SMF.
6. Whenever SMF sends another SM Create request to SM service, SM service sends a request to NRF client to fetch the UDR profiles.
7. As the data is already cached, NRF client fetches the UDR profiles with the matching nfsetID from its cache and responds to SM service.
8. After the successful creation of SMPolicyAssociation, SM service responds to SMF.

Managing the Feature

Enable

By default, this feature is disabled. You can enable caching of the NF Profiles at NRF client using CNC Console or REST API for Policy.

Enable using CNC Console:

To enable the feature using CNC Console configure:

- **Force Discovery** parameter under **On Demand Discovery Caching** section on **NF Communication Profile** page. The **NF Communication Profile** page is available under **Common Data** for **Service Configurations**.
- **Enable Caching** parameter under **On Demand Discovery Caching** section on **NRF Agent** page, The **NRF Agent** page is available under **Service Configurations**.

For more information about enabling the feature through CNC Console, see NF Communication Profiles.

Enable using REST API:

You can enable the feature using:

- NF Communication Profiles API by configuring the value of `forceDiscovery` paramter under `onDemandDiscoveryCaching` section to 0.
- NRF Agent Service API by configuring the value of `enableFeature` parameter under `onDemandDiscoveryCaching` section to `true`.

For more information about enabling the feature through REST API, see *NF Communication Profiles* in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

You can configure this feature using the CNC Console or REST API for Policy.

Configure using CNC Console

- To configure caching of the NF profiles at NRF client at global level, perform the following configurations under **On Demand Discovery Caching** section on **NRF Agent** page, The **NRF Agent** page is available under **Service Configurations**. For more information, see NRF Agent.

- To configure the retention period of the cached data, perform the following configurations under **On Demand Discovery Caching** section on **NF Communication Profile** page. The **NF Communication Profile** page is available under **Common Data** for **Service Configurations**. For more information, see NF Communication Profiles.

Configure using REST API

You can configure this feature using NF Communication Profiles and NRF Agent Service APIs.

For more information, see *NF Communication Profiles* and *NRF Agent* sections in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Observability

Metrics

The following NRF Client metrics are used for this feature:

- occnp_nrfclient_discovery_cache_support_force_discovery_total
- occnp_nrfclient_discovery_cache_support_cache_hit_total
- occnp_nrfclient_discovery_cache_support_about_to_expire_total
- occnp_nrfclient_discovery_cache_support_expired_total
- occnp_nrfclient.discovery.cache.support.failover
- occnp_nrfclient_discovery_cache_support_cache_non_cache_total
- occnp_nrfclient_discovery_cache_support_empty_response_total
- occnp_nrfclient_discovery_cache_support_cache_lookup_seconds
- occnp_nrfclient_discovery_cache_support_cache_lookup_seconds_bucket
- occnp_nrfclient_discovery_cache_support_cache_lookup_seconds_count
- occnp_nrfclient_discovery_cache_support_cache_lookup_seconds_max
- occnp_nrfclient_discovery_cache_support_cache_lookup_seconds_sum
- occnp_nrfclient_discovery_cache_support_backend_response_seconds
- occnp_nrfclient_discovery_cache_support_backend_response_seconds_bucket
- occnp_nrfclient_discovery_cache_support_backend_response_seconds_count
- occnp_nrfclient_discovery_cache_support_backend_response_seconds_max
- occnp_nrfclient_discovery_cache_support_backend_response_seconds_sum
- occnp_nrfclient_discovery_cache_support_profiles_bucket
- occnp_nrfclient_discovery_cache_support_profiles_count
- occnp_nrfclient_discovery_cache_support_profiles_max
- occnp_nrfclient_discovery_cache_support_profiles_sum

For more information, see NRF Client Metrics.

Logging

PCF services

```
AM-Service:
{"instant":
{"epochSecond":1687351821,"nanoOfSecond":727462057},"thread":"HttpLoggingJetty
```

HttpClient@4be14ed1-134","level":"DEBUG","loggerName":"ocpm.pcf.service.am.retry.AlternateRoutingHelper","message":"nfcommunication profile attached in settings NFCommProfile{nfCommunicationProfileName='nfcp_rtp_sameLoc', policyNfCommunicationModel=MODEL_C(3), nfDiscoverySettings=NFDiscoverySettings{discoveryParameters=[ocpm.pcf.resource.common.model.DiscoveryParamCfg@33ee9d8b], sendDiscoveryHeaderInitMsg=false, sendDiscoveryHeaderSubsequentMsg=false, sendTargetApiRootHeaderInitMsg=true, onDemandDiscoverySettings=OnDemandDiscoveryCaching{forceDiscovery=1, retentionPeriod=null}}, nfBindingSettings=NFBindingSettings{cfgBindingLevel=NF_SET(1), sendBindingHeader=true, sendRoutingBindingHeader=true, sendCallbackHeader=true}, nfServerSettings=NFServerSettings{sendServerHeader=false, serverHeaderErrorCodes=null}, retryAndAlternateRoutingSettings=ocpm.pcf.resource.common.model.RetryAndAlternateRoutingSettings@4a752f56, nfCorrelationSettings=null}","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":134,"threadPriority":5,"messageTimestamp":"2023-06-21T12:50:21.727+0000"}
{"instant":
{"epochSecond":1687351821,"nanoOfSecond":727672162},"thread":"HttpLoggingJettyHttpClient@4be14ed1-134","level":"DEBUG","loggerName":"ocpm.pcf.service.am.retry.AlternateRoutingHelper","message":"ocForcedRediscovery is 1 and retention Period is
null","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":134,"threadPriority":5,"messageTimestamp":"2023-06-21T12:50:21.727+0000"}
SM-Service:

{"instant":
{"epochSecond":1694159028,"nanoOfSecond":405961795},"thread":"boundedElastic-4","level":"DEBUG","loggerName":"ocpm.pcf.service.sm.domain.component.retry.profiles.AlternateRoutingHelper","message":" ocForcedRediscovery is 1 and retentionPeriod is 5000
","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":210,"threadPriority":5,"messageTimestamp":"2023-09-08T07:43:48.405+0000"
}

UE-Service:
{"instant":
{"epochSecond":1694169865,"nanoOfSecond":366771034},"thread":"pool-3-thread-10","level":"DEBUG","loggerName":"ocpm.pcf.service.uepolicy.routing.AlternateRoutingHelper","message":"NF Communication Profile attached in service configuration is NFCommProfile{nfCommunicationProfileName='nfcp_rtp', policyNfCommunicationModel=MODEL_C(3), nfDiscoverySettings=NFDiscoverySettings{discoveryParameters=[ocpm.pcf.resource.common.model.DiscoveryParamCfg@5d2c8a02], sendDiscoveryHeaderInitMsg=false, sendDiscoveryHeaderSubsequentMsg=false, sendTargetApiRootHeaderInitMsg=true}, onDemandDiscoveryCaching=OnDemandDiscoveryCaching{forceDiscovery=1, retentionPeriod=5000}, nfBindingSettings=NFBindingSettings{cfgBindingLevel=NF_SET(1), sendBindingHeader=true, sendRoutingBindingHeader=true, sendCallbackHeader=true, sendServiceName=false}, nfServerSettings=NFServerSettings{sendServerHeader=false, serverHeaderErrorCodes=null}, retryAndAlternateRoutingSettings=ocpm.pcf.resource.common.model.RetryAndAltern

ateRoutingSettings@468b7e0a,
nfCorrelationSettings=null}","endOfBatch":false,"loggerFqcn":"org.apache.loggi
ng.slf4j.Log4jLogger","threadId":890,"threadPriority":5,"messageTimestamp":"20
23-09-08T10:44:25.366+0000"}
{"instant":
{"epochSecond":1694169865,"nanoOfSecond":366943366},"thread":"pool-3-
thread-10","level":"DEBUG","loggerName":"ocpm.pcf.service.uepolicy.routing.Alt
ernateRoutingHelper","message":" ocForcedRediscovery is 1 and retentionPeriod
is 5000
","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","thre
adId":890,"threadPriority":5,"messageTimestamp":"2023-09-08T10:44:25.366+0000"
}

User-Service:

{"instant":
{"epochSecond":1694170047,"nanoOfSecond":943574493},"thread":"UserService_Thre
adPool_16","level":"DEBUG","loggerName":"ocpm.pcf.service.ud.dbplugin.ds.Alter
nateRouteServiceHelper","message":"NF-Communication Profile is
NFCommProfile{nfCommunicationProfileName='nfcp_rtp_diffSet_sameLoc',
policyNfCommunicationModel=MODEL_C(3),
nfDiscoverySettings=NFDiscoverySettings{discoveryParameters=[ocpm.pcf.resource
.common.model.DiscoveryParamCfg@75f2f60d], sendDiscoveryHeaderInitMsg=false,
sendDiscoveryHeaderSubsequentMsg=false, sendTargetApiRootHeaderInitMsg=true},
onDemandDiscoveryCaching=OnDemandDiscoveryCaching{forceDiscovery=1,
retentionPeriod=5000},
nfBindingSettings=NFBindingSettings{cfgBindingLevel=NF_SET(1),
sendBindingHeader=true, sendRoutingBindingHeader=true,
sendCallbackHeader=true, sendServiceName=false},
nfServerSettings=NFServerSettings{sendServerHeader=false,
serverHeaderErrorCodes=null},
retryAndAlternateRoutingSettings=ocpm.pcf.resource.common.model.RetryAndAltern
ateRoutingSettings@3b6541ce, nfCorrelationSettings=null}, oc-forced-discovery
is 1 and retention period is
5000","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","
threadId":276,"threadPriority":5,"messageTimestamp":"2023-09-08T10:47:27.943+0
000"}
{"instant":
{"epochSecond":1694170047,"nanoOfSecond":943677402},"thread":"UserService_Thre
adPool_16","level":"DEBUG","loggerName":"ocpm.pcf.service.ud.dbplugin.ds.Alter
nateRouteServiceHelper","message":" OC-Forced-Rediscovery is
1","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","thr
eadId":276,"threadPriority":5,"messageTimestamp":"2023-09-08T10:47:27.943+0000
"}

NRF Client

For SM Service:

{"instant":
{"epochSecond":1702446905,"nanoOfSecond":450166996},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryRule","message":"Checking if cache applies for a discovery
request (DiscoveryCache feature enabled: true, OC-Force-Rediscovery
requested:
false)","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger"

```
,"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T05:55:05.450+
0000"}
{"instant":
{"epochSecond":1702446905,"nanoOfSecond":507483619},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryRule","message":"Query Parameters validation. IsFullyMatch ->
true, Params from request -> [target-nf-type, requester-nf-type, service-
names, target-nf-set-id], Params from config -> [target-nf-type, requester-nf-
type, service-names, target-nf-set-
id]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","t
hreadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T05:55:05.507+000
0"}
{"instant":
{"epochSecond":1702446905,"nanoOfSecond":507762621},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryrul
es.DiscoveryRule","message":"DiscoveryCache is enabled and queryParameters
matched. Proceeding with DiscoveryResponse retrieval from
memory","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger"
,"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T05:55:05.507+
0000"}
{"instant":
{"epochSecond":1702446905,"nanoOfSecond":508175763},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryEvaluator","message":"Rule applied:
CacheRule","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogg
er","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T05:55:05.5
08+0000"}
{"instant":
{"epochSecond":1702446905,"nanoOfSecond":508393875},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.CacheRule","message":"Applying Cache Rule, grabbing NRF response from
Cache","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger",
"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T05:55:05.508+0
000"}
{"instant":
{"epochSecond":1702446905,"nanoOfSecond":813463474},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Unsuccessful retrieval of
Discovery Response from Cache/DB, proceeding retrieval from NRF and saving
response in Cache/
DB","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","th
readId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T05:55:05.813+0000
"}
{"instant":
{"epochSecond":1702446905,"nanoOfSecond":813770323},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.nrf
.GetNrfDiscoveryResponseUseCase","message":"SearchData :
{\"targetNfType\":\"SMF\",\"serviceNames\":[\"nsmf-
pdusession\"],\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":false,\"serv
iceInfoParamsPresent\":true,\"enableF3\":true,\"enableF5\":false,\"targetNfSet
Id\":\"set001.smfset.5gc.mnc012.mcc345\",\"retentionPeriod\":5000,\"rawQueryPa
rameters\":\"target-nf-type=SMF&requester-nf-type=PCF&target-nf-set-
id=set001.smfset.5gc.mnc012.mcc345&service-names=nsmf-
pdusession\",\"forceRediscoveryEnabled\":false}","endOfBatch":false,"loggerFqc
n":"org.apache.logging.slf4j.Log4jLogger","threadId":90,"threadPriority":5,"me
ssageTimestamp":"2023-12-13T05:55:05.813+0000"}
```

{"instant":
{"epochSecond":1702446905,"nanoOfSecond":814870367},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Entering sendOnDemandNfDiscoverRequest
function","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogge
r","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T05:55:05.8
14+0000"}
{"instant":
{"epochSecond":1702446905,"nanoOfSecond":819222220},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Send NfDiscover request with searchData:
{\"targetNfType\":\"SMF\",\"serviceNames\":[\"nsmf-
pdusession\"],\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":false,\"serv
iceInfoParamsPresent\":true,\"enableF3\":true,\"enableF5\":false,\"targetNfSet
Id\":\"set001.smfset.5gc.mnc012.mcc345\",\"retentionPeriod\":5000,\"rawQueryPa
rameters\":\"target-nf-type=SMF&requester-nf-type=PCF&target-nf-set-
id=set001.smfset.5gc.mnc012.mcc345&service-names=nsmf-
pdusession\",\"forceRediscoveryEnabled\":false}","endOfBatch":false,"loggerFqc
n":"org.apache.logging.slf4j.Log4jLogger","threadId":101,"threadPriority":5,"m
essageTimestamp":"2023-12-13T05:55:05.819+0000"}
{"instant":
{"epochSecond":1702446905,"nanoOfSecond":819761612},"thread":"pool-10-
thread-1","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientA
pi","message":"sendRequestToNrf nrfRequest : NRFRequest [scheme=http,
rootURI=nf1stub.gi-dina.svc:8080, resourcePath=/nnrf-disc/v1/nf-instances,
method=GET, body=null, headers=[],
searchData={\"targetNfType\":\"SMF\",\"serviceNames\":[\"nsmf-
pdusession\"],\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":false,\"serv
iceInfoParamsPresent\":true,\"enableF3\":true,\"enableF5\":false,\"targetNfSet
Id\":\"set001.smfset.5gc.mnc012.mcc345\",\"retentionPeriod\":5000,\"rawQueryPa
rameters\":\"target-nf-type=SMF&requester-nf-type=PCF&target-nf-set-
id=set001.smfset.5gc.mnc012.mcc345&service-names=nsmf-
pdusession\",\"forceRediscoveryEnabled\":false}, routeCount=0,
requestType=NFDISCOVER]","endOfBatch":false,"loggerFqcn":"org.apache.logging.s
lf4j.Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-1
2-13T05:55:05.819+0000"}
{"instant":
{"epochSecond":1702446905,"nanoOfSecond":823597913},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Returning retryConfig for ALL_REQUESTS NRFRetryConfig
[serviceRequestType=ALL_REQUESTS, primaryNRFRetryCount=0,
nonPrimaryNRFRetryCount=0, alternateNRFRetryCount=-1,
errorReasonsForFailure=[503, 504, 500, SocketTimeoutException,
JsonProcessingException, UnknownHostException, NoRouteToHostException],
gatewayErrorCodes=[503],
requestTimeout=10]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.
Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T
05:55:05.823+0000"}
{"instant":
{"epochSecond":1702446905,"nanoOfSecond":824455718},"thread":"pool-10-
thread-1","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientA
pi","message":"Sending request to NRF nf1stub.gi-dina.svc:8080, routeCount=0,
attempt=0","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogg
er","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T05:55:05.
824+0000"}
{"instant":

{"epochSecond":1702446905,"nanoOfSecond":824609434},"thread":"pool-10-thread-1","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientApi","message":"trigger","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T05:55:05.824+0000"}
{"instant":
{"epochSecond":1702446905,"nanoOfSecond":824752596},"thread":"pool-10-thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientApi","message":"Returning retryConfig for ALL_REQUESTS NRFRetryConfig [serviceRequestType=ALL_REQUESTS, primaryNRFRetryCount=0, nonPrimaryNRFRetryCount=0, alternateNRFRetryCount=-1, errorReasonsForFailure=[503, 504, 500, SocketTimeoutException, JsonProcessingException, UnknownHostException, NoRouteToHostException], gatewayErrorCodes=[503], requestTimeout=10]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T05:55:05.824+0000"}
{"instant":
{"epochSecond":1702446905,"nanoOfSecond":928605343},"thread":"pool-10-thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientApi","message":"Request returned response with status code : 200","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T05:55:05.928+0000"}
{"instant":
{"epochSecond":1702446906,"nanoOfSecond":28211509},"thread":"pool-10-thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientApi","message":"Successful Response code received","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T05:55:06.028+0000"}
{"instant":
{"epochSecond":1702446906,"nanoOfSecond":28712480},"thread":"pool-10-thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientApi","message":"sendOnDemandNfDiscoverRequest with searchData: {\"targetNfType\":\"SMF\",\"serviceNames\":[\"nsmf-pdusession\"],\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":false,\"serviceInfoParamsPresent\":true,\"enableF3\":true,\"enableF5\":false,\"targetNfSetId\":\"set001.smfset.5gc.mnc012.mcc345\",\"retentionPeriod\":5000,\"...

AM Service:

Discovery of AMF for the first time:

{"instant":
{"epochSecond":1702447453,"nanoOfSecond":267776264},"thread":"XNIO-1 task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryrules.DiscoveryRule","message":"Checking if cache applies for a discovery request (DiscoveryCache feature enabled: true, OC-Force-Rediscovery requested: false)","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:13.267+0000"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":267885332},"thread":"XNIO-1

task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryRule","message":"Query Parameters validation. IsFullyMatch ->
true, Params from request -> [target-nf-type, requester-nf-type, target-nf-
set-id], Params from config -> [target-nf-type, requester-nf-type, service-
names, target-nf-set-
id]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","t
hreadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:13.267+000
0"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":268118684},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryrul
es.DiscoveryRule","message":"DiscoveryCache is enabled and queryParameters
matched. Proceeding with DiscoveryResponse retrieval from
memory","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger"
,"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:13.268+
0000"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":268173836},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryEvaluator","message":"Rule applied:
CacheRule","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogg
er","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:13.2
68+0000"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":268214963},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.CacheRule","message":"Applying Cache Rule, grabbing NRF response from
Cache","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger",
"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:13.268+0
000"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":279713217},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Unsuccessful retrieval of
Discovery Response from Cache/DB, proceeding retrieval from NRF and saving
response in Cache/
DB","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","th
readId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:13.279+0000
"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":279876856},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.nrf
.GetNrfDiscoveryResponseUseCase","message":"SearchData :
{\"targetNfType\":\"AMF\",\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":
false,\"serviceInfoParamsPresent\":false,\"enableF3\":true,\"enableF5\":false,
\"targetNfSetId\":\"set001.region01.amfset.5gc.mnc012.mcc345\",\"retentionPeri
od\":3000,\"rawQueryParameters\":\"target-nf-type=AMF&requester-nf-
type=PCF&target-nf-set-
id=set001.region01.amfset.5gc.mnc012.mcc345\",\"forceRediscoveryEnabled\":fals
e}","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","th
readId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:13.279+0000
"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":280256953},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Entering sendOnDemandNfDiscoverRequest

function","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogge
r","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:13.2
80+0000"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":282059938},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Send NfDiscover request with searchData:
{\"targetNfType\":\"AMF\",\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":
false,\"serviceInfoParamsPresent\":false,\"enableF3\":true,\"enableF5\":false,
\"targetNfSetId\":\"set001.region01.amfset.5gc.mnc012.mcc345\",\"retentionPeri
od\":3000,\"rawQueryParameters\":\"target-nf-type=AMF&requester-nf-
type=PCF&target-nf-set-
id=set001.region01.amfset.5gc.mnc012.mcc345\",\"forceRediscoveryEnabled\":fals
e}","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","th
readId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:13.282+000
0"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":282199870},"thread":"pool-10-
thread-1","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientA
pi","message":"sendRequestToNrf nrfRequest : NRFRequest [scheme=http,
rootURI=nf1stub.gi-dina.svc:8080, resourcePath=/nnrf-disc/v1/nf-instances,
method=GET, body=null, headers=[],
searchData={\"targetNfType\":\"AMF\",\"requesterNfType\":\"PCF\",\"nfInfoParam
sPresent\":false,\"serviceInfoParamsPresent\":false,\"enableF3\":true,\"enable
F5\":false,\"targetNfSetId\":\"set001.region01.amfset.5gc.mnc012.mcc345\",\"re
tentionPeriod\":3000,\"rawQueryParameters\":\"target-nf-type=AMF&requester-nf-
type=PCF&target-nf-set-
id=set001.region01.amfset.5gc.mnc012.mcc345\",\"forceRediscoveryEnabled\":fals
e}, routeCount=0,
requestType=NFDISCOVER]","endOfBatch":false,"loggerFqcn":"org.apache.logging.s
lf4j.Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-1
2-13T06:04:13.282+0000"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":282424344},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Returning retryConfig for ALL_REQUESTS NRFRetryConfig
[serviceRequestType=ALL_REQUESTS, primaryNRFRetryCount=0,
nonPrimaryNRFRetryCount=0, alternateNRFRetryCount=-1,
errorReasonsForFailure=[503, 504, 500, SocketTimeoutException,
JsonProcessingException, UnknownHostException, NoRouteToHostException],
gatewayErrorCodes=[503],
requestTimeout=10]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.
Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T
06:04:13.282+0000"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":282532830},"thread":"pool-10-
thread-1","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientA
pi","message":"Sending request to NRF nf1stub.gi-dina.svc:8080, routeCount=0,
attempt=0","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogg
er","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:13.
282+0000"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":282579698},"thread":"pool-10-
thread-1","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientA
pi","message":"trigger","endOfBatch":false,"loggerFqcn":"org.apache.logging.sl
f4j.Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12

-13T06:04:13.282+0000"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":282628294},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Returning retryConfig for ALL_REQUESTS NRFRetryConfig
[serviceRequestType=ALL_REQUESTS, primaryNRFRetryCount=0,
nonPrimaryNRFRetryCount=0, alternateNRFRetryCount=-1,
errorReasonsForFailure=[503, 504, 500, SocketTimeoutException,
JsonProcessingException, UnknownHostException, NoRouteToHostException],
gatewayErrorCodes=[503],
requestTimeout=10]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.
Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T
06:04:13.282+0000"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":306471240},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Request returned response with status code :
200","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","t
hreadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:13.306+00
00"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":311674972},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Successful Response code
received","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogge
r","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:13.3
11+0000"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":311771862},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"sendOnDemandNfDiscoverRequest with searchData:
{\"targetNfType\":\"AMF\",\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":
false,\"serviceInfoParamsPresent\":false,\"enableF3\":true,\"enableF5\":false,
\"targetNfSetId\":\"set001.region01.amfset.5gc.mnc012.mcc345\",\"retentionPeri
od\":3000,\"rawQueryParameters\":\"target-nf-type=AMF&requester-nf-
type=PCF&target-nf-set-
id=set001.region01.amfset.5gc.mnc012.mcc345\",\"forceRediscoveryEnabled\":fals
e} returned status code :
200","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","t
hreadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:13.311+00
00"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":311992401},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.nrf
.GetNrfDiscoveryResponseUseCase","message":"Received Discovery Result. Code:
200, Body (SearchResult): {\"validityPeriod\": 300, \"nfInstances\":
[{\"nfInstanceId\": \"13515195-c537-4645-9b97-96ec797faaaf\",
\"nfSetIdList\": [\"set001.region01.amfset.5gc.mnc012.mcc345\"],
\"allowedNfTypes\": [\"PCF\", \"SMF\"], \"nfType\": \"AMF\", \"plmnList\":
[{\"mcc\": \"450\", \"mnc\": \"05\"}], \"nfStatus\": \"REGISTERED\",
\"fqdn\": \"nf2stub.gi-dina.svc\", \"priority\": 1, \"capacity\": 100,
\"load\": 50, \"locality\": \"1\", \"amfInfo\": {\"amfRegionId\": \"01\",
\"amfSetId\": \"001\", \"guamiList\": [{\"plmnId\": {\"mcc\": \"450\",
\"mnc\": \"05\"}, \"amfId\": \"010041\"}]}, \"nfServices\":
[{\"serviceInstanceId\": \"aaaa-bbbb-cccc-dddd\", \"serviceName\": \"namf-
comm\", \"versions\": [{\"apiVersionInUri\": \"v1\", \"apiFullVersion\":

\"1.1.0\", \"expiry\": \"2020-07-30T12:09:55.65Z\"}], \"scheme\": \"http\",
\"nfServiceStatus\": \"REGISTERED\", \"ipEndPoints\": [{\"ipv4Address\":
null, \"ipv6Address\": null, \"transport\": \"TCP\", \"port\": 8080}],
\"allowedPlmns\": [{\"mcc\": \"450\", \"mnc\": \"05\"}]}]},
{\"nfInstanceId\": \"13515195-c537-4645-9b97-96ec797fbbbf\", \"nfSetIdList\":
[\"set001.region01.amfset.5gc.mnc012.mcc345\"], \"allowedNfTypes\": [\"PCF\",
\"SMF\"], \"nfType\": \"AMF\", \"plmnList\": [{\"mcc\": \"450\", \"mnc\":
\"05\"}], \"nfStatus\": \"REGISTERED\", \"fqdn\": \"nf21stub.gi-dina.svc\",
\"priority\": 2, \"capacity\": 100, \"load\": 50, \"locality\": \"1\",
\"amfInfo\": {\"amfRegionId\": \"01\", \"amfSetId\": \"001\", \"guamiList\":
[{\"plmnId\": {\"mcc\": \"450\", \"mnc\": \"05\"}, \"amfId\": \"010041\"}]},
\"nfServices\": [{\"serviceInstanceId\": \"aaaa-bbbb-cccc-dddd\",
\"serviceName\": \"namf-comm\", \"versions\": [{\"apiVersionInUri\": \"v1\",
\"apiFullVersion\": \"1.1.0\", \"expiry\": \"2020-07-30T12:09:55.65Z\"}],
\"scheme\": \"http\", \"nfServiceStatus\": \"REGISTERED\", \"ipEndPoints\":
[{\"ipv4Address\": null, \"ipv6Address\": null, \"transport\": \"TCP\",
\"port\": 8080}], \"allowedPlmns\": [{\"mcc\": \"450\", \"mnc\": \"05\"}]}]},
{\"nfInstanceId\": \"13515195-c537-4645-9b97-96ec797fcccf\", \"nfSetIdList\":
[\"set001.region01.amfset.5gc.mnc012.mcc345\"], \"allowedNfTypes\": [\"PCF\",
\"SMF\"], \"nfType\": \"AMF\", \"plmnList\": [{\"mcc\": \"450\", \"mnc\":
\"05\"}], \"nfStatus\": \"REGISTERED\", \"fqdn\": \"nf3stub.gi-dina.svc\",
\"priority\": 4, \"capacity\": 100, \"load\": 60, \"locality\": \"2\",
\"amfInfo\": {\"amfRegionId\": \"01\", \"amfSetId\": \"001\", \"guamiList\":
[{\"plmnId\": {\"mcc\": \"450\", \"mnc\": \"05\"}, \"amfId\": \"010041\"}]},
\"nfServices\": [{\"serviceInstanceId\": \"aaaa-bbbb-cccc-dddd\",
\"serviceName\": \"namf-comm\", \"versions\": [{\"apiVersionInUri\": \"v1\",
\"apiFullVersion\": \"1.1.0\", \"expiry\": \"2020-07-30T12:09:55.65Z\"}],
\"scheme\": \"http\", \"nfServiceStatus\": \"REGISTERED\", \"ipEndPoints\":
[{\"ipv4Address\": null, \"ipv6Address\": null, \"transport\": \"TCP\",
\"port\": 8080}], \"allowedPlmns\": [{\"mcc\": \"450\", \"mnc\": \"05\"}]}]},
{\"nfInstanceId\": \"13515195-c537-4645-9b97-96ec797fdddf\", \"nfSetIdList\":
[\"set001.region01.amfset.5gc.mnc012.mcc345\"], \"allowedNfTypes\": [\"PCF\",
\"SMF\"], \"nfType\": \"AMF\", \"plmnList\": [{\"mcc\": \"450\", \"mnc\":
\"05\"}], \"nfStatus\": \"REGISTERED\", \"fqdn\": \"nf31stub.gi-dina.svc\",
\"priority\": 2, \"capacity\": 100, \"load\": 50, \"locality\": \"2\",
\"amfInfo\": {\"amfRegionId\": \"01\", \"amfSetId\": \"001\", \"guamiList\":
[{\"plmnId\": {\"mcc\": \"450\", \"mnc\": \"05\"}, \"amfId\": \"010041\"}]},
\"nfServices\": [{\"serviceInstanceId\": \"aaaa-bbbb-cccc-dddd\",
\"serviceName\": \"namf-comm\", \"versions\": [{\"apiVersionInUri\": \"v1\",
\"apiFullVersion\": \"1.1.0\", \"expiry\": \"2020-07-30T12:09:55.65Z\"}],
\"scheme\": \"http\", \"nfServiceStatus\": \"REGISTERED\", \"ipEndPoints\":
[{\"ipv4Address\": null, \"ipv6Address\": null, \"transport\": \"TCP\",
\"port\": 8080}], \"allowedPlmns\": [{\"mcc\": \"450\", \"mnc\": \"05\"}]}]},
{\"nfInstanceId\": \"13515195-c537-4645-9b97-96ec797feeef\", \"nfSetIdList\":
[\"set001.region01.amfset.5gc.mnc012.mcc345\"], \"allowedNfTypes\": [\"PCF\",
\"SMF\"], \"nfType\": \"AMF\", \"plmnList\": [{\"mcc\": \"450\", \"mnc\":
\"05\"}], \"nfStatus\": \"REGISTERED\", \"fqdn\": \"nf32stub.gi-dina.svc\",
\"priority\": 5, \"capacity\": 100, \"load\": 60, \"locality\": \"2\",
\"amfInfo\": {\"amfRegionId\": \"01\", \"amfSetId\": \"001\", \"guamiList\":
[{\"plmnId\": {\"mcc\": \"450\", \"mnc\": \"05\"}, \"amfId\": \"010041\"}]},
\"nfServices\": [{\"serviceInstanceId\": \"aaaa-bbbb-cccc-dddd\",
\"serviceName\": \"namf-comm\", \"versions\": [{\"apiVersionInUri\": \"v1\",
\"apiFullVersion\": \"1.1.0\", \"expiry\": \"2020-07-30T12:09:55.65Z\"}],
\"scheme\": \"http\", \"nfServiceStatus\": \"REGISTERED\", \"ipEndPoints\":
[{\"ipv4Address\": null, \"ipv6Address\": null, \"transport\": \"TCP\",
\"port\": 8080}], \"allowedPlmns\": [{\"mcc\": \"450\", \"mnc\":

\"05\"}]}]}]}","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4j
Logger","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:
13.311+0000"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":313001061},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.nrf
.GetNrfDiscoveryResponseUseCase","message":"Dynamic discovery cache enabled :
false","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger",
"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:13.313+0
000"}
{"instant":
{"epochSecond":1702447453,"nanoOfSecond":353052224},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Discovery Response fetched :
DiscoverySearchResult{queryParams='target-nf-type=AMF&requester-nf-
type=PCF&target-nf-set-id=set001.region01.amfset.5gc.mnc012.mcc345',
searchResult={\"validityPeriod\":300,\"nfInstances\":
[{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797faaaf\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf2stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":1,\"capacity\":100,\"load\":50,\"locality\":\"1
\",\"nfSetIdList\":[\"set001.region01.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}]}]},{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797fbbbf\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf21stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":2,\"capacity\":100,\"load\":50,\"locality\":\"1
\",\"nfSetIdList\":[\"set001.region01.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}]}]},{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797fcccf\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf3stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":4,\"capacity\":100,\"load\":60,\"locality\":\"2
\",\"nfSetIdList\":[\"set001.region01.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":

[{\"mcc\":\"450\",\"mnc\":\"05\"}]}]},{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797fdddf\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf31stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":2,\"capacity\":100,\"load\":50,\"locality\":\"2
\",\"nfSetIdList\":[\"set001.region01.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}]}]},{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797feeef\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf32stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":5,\"capacity\":100,\"load\":60,\"locality\":\"2
\",\"nfSetIdList\":[\"set001.region01.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}]}]}],\"nrfSupportedFeatures\":\"72\"},
sourceType=NRF}","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log
4jLogger","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:0
4:13.353+0000"}

Discovery of AMF for the second time

{"instant":
{"epochSecond":1702447495,"nanoOfSecond":560602127},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryRule","message":"Checking if cache applies for a discovery
request (DiscoveryCache feature enabled: true, OC-Force-Rediscovery
requested:
false)","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger"
,"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:55.560+
0000"}
{"instant":
{"epochSecond":1702447495,"nanoOfSecond":560665276},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryRule","message":"Query Parameters validation. IsFullyMatch ->
true, Params from request -> [target-nf-type, requester-nf-type, target-nf-
set-id], Params from config -> [target-nf-type, requester-nf-type, service-
names, target-nf-set-
id]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","t
hreadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:55.560+000
0"}
{"instant":

{"epochSecond":1702447495,"nanoOfSecond":560736605},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryrul
es.DiscoveryRule","message":"DiscoveryCache is enabled and queryParameters
matched. Proceeding with DiscoveryResponse retrieval from
memory","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger"
,"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:55.560+
0000"}
{"instant":
{"epochSecond":1702447495,"nanoOfSecond":560773857},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryEvaluator","message":"Rule applied:
CacheRule","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogg
er","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:55.5
60+0000"}
{"instant":
{"epochSecond":1702447495,"nanoOfSecond":560806304},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.CacheRule","message":"Applying Cache Rule, grabbing NRF response from
Cache","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger",
"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:55.560+0
000"}
{"instant":
{"epochSecond":1702447495,"nanoOfSecond":578396526},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Successful retrieval of
Discovery Response stored in Cache/DB, proceeding with
validation","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLog
ger","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:55.
578+0000"}
{"instant":
{"epochSecond":1702447495,"nanoOfSecond":578450087},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Discovery Response retrieved
from Cache/DB isAboutToExpire value : false
","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","thre
adId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:55.578+0000"}
{"instant":
{"epochSecond":1702447495,"nanoOfSecond":578490288},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Discovery Response retrieved
from Cache/DB is expired value : false
","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","thre
adId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:04:55.578+0000"}
{"instant":
{"epochSecond":1702447495,"nanoOfSecond":589440279},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Discovery Response fetched :
DiscoverySearchResult{queryParams='target-nf-type=AMF&requester-nf-
type=PCF&target-nf-set-id=set001.region01.amfset.5gc.mnc012.mcc345',
searchResult={\"validityPeriod\":300,\"nfInstances\":
[{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797faaaf\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf2stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":1,\"capacity\":100,\"load\":50,\"locality\":\"1
\",\"nfSetIdList\":[\"set001.region01.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":

{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}]}]},{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797fbbbf\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf21stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":2,\"capacity\":100,\"load\":50,\"locality\":\"1
\",\"nfSetIdList\":[\"set001.region01.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}]}]},{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797fcccf\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf3stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":4,\"capacity\":100,\"load\":60,\"locality\":\"2
\",\"nfSetIdList\":[\"set001.region01.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}]}]},{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797fdddf\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf31stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":2,\"capacity\":100,\"load\":50,\"locality\":\"2
\",\"nfSetIdList\":[\"set001.region01.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}]}]},{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797feeef\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf32stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":5,\"capacity\":100,\"load\":60,\"locality\":\"2
\",\"nfSetIdList\":[\"set001.region01.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":

[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}]}]}],\"nrfSupportedFeatures\":\"72\"},
sourceType=CACHE}","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.L
og4jLogger","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06
:04:55.589+0000"}

UE Policy Service:

Discovery of AMF for UE policy Association for the first time:

{"instant":
{"epochSecond":1702447646,"nanoOfSecond":16500495},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryRule","message":"Checking if cache applies for a discovery
request (DiscoveryCache feature enabled: true, OC-Force-Rediscovery
requested:
false)","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger"
,"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:07:26.016+
0000"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":16566854},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryRule","message":"Query Parameters validation. IsFullyMatch ->
true, Params from request -> [target-nf-type, requester-nf-type, target-nf-
set-id], Params from config -> [target-nf-type, requester-nf-type, target-nf-
set-
id]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","t
hreadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:07:26.016+000
0"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":16634530},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryrul
es.DiscoveryRule","message":"DiscoveryCache is enabled and queryParameters
matched. Proceeding with DiscoveryResponse retrieval from
memory","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger"
,"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:07:26.016+
0000"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":16668622},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryEvaluator","message":"Rule applied:
CacheRule","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogg
er","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:07:26.0
16+0000"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":16697142},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.CacheRule","message":"Applying Cache Rule, grabbing NRF response from
Cache","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger",
"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:07:26.016+0
000"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":26184667},"thread":"XNIO-1

task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Unsuccessful retrieval of
Discovery Response from Cache/DB, proceeding retrieval from NRF and saving
response in Cache/
DB","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","th
readId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:07:26.026+0000
"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":26250918},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.nrf
.GetNrfDiscoveryResponseUseCase","message":"SearchData :
{\"targetNfType\":\"AMF\",\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":
false,\"serviceInfoParamsPresent\":false,\"enableF3\":true,\"enableF5\":false,
\"targetNfSetId\":\"set001.region48.amfset.5gc.mnc012.mcc345\",\"retentionPeri
od\":5000,\"rawQueryParameters\":\"target-nf-type=AMF&requester-nf-
type=PCF&target-nf-set-
id=set001.region48.amfset.5gc.mnc012.mcc345\",\"forceRediscoveryEnabled\":fals
e}","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","th
readId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:07:26.026+0000
"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":26465561},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Entering sendOnDemandNfDiscoverRequest
function","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogge
r","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:07:26.0
26+0000"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":27251963},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Send NfDiscover request with searchData:
{\"targetNfType\":\"AMF\",\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":
false,\"serviceInfoParamsPresent\":false,\"enableF3\":true,\"enableF5\":false,
\"targetNfSetId\":\"set001.region48.amfset.5gc.mnc012.mcc345\",\"retentionPeri
od\":5000,\"rawQueryParameters\":\"target-nf-type=AMF&requester-nf-
type=PCF&target-nf-set-
id=set001.region48.amfset.5gc.mnc012.mcc345\",\"forceRediscoveryEnabled\":fals
e}","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","th
readId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:07:26.027+000
0"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":27393966},"thread":"pool-10-
thread-1","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientA
pi","message":"sendRequestToNrf nrfRequest : NRFRequest [scheme=http,
rootURI=nf1stub.gi-dina.svc:8080, resourcePath=/nnrf-disc/v1/nf-instances,
method=GET, body=null, headers=[],
searchData={\"targetNfType\":\"AMF\",\"requesterNfType\":\"PCF\",\"nfInfoParam
sPresent\":false,\"serviceInfoParamsPresent\":false,\"enableF3\":true,\"enable
F5\":false,\"targetNfSetId\":\"set001.region48.amfset.5gc.mnc012.mcc345\",\"re
tentionPeriod\":5000,\"rawQueryParameters\":\"target-nf-type=AMF&requester-nf-
type=PCF&target-nf-set-
id=set001.region48.amfset.5gc.mnc012.mcc345\",\"forceRediscoveryEnabled\":fals
e}, routeCount=0,
requestType=NFDISCOVER]","endOfBatch":false,"loggerFqcn":"org.apache.logging.s
lf4j.Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-1
2-13T06:07:26.027+0000"}

{"instant":
{"epochSecond":1702447646,"nanoOfSecond":27518993},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Returning retryConfig for ALL_REQUESTS NRFRetryConfig
[serviceRequestType=ALL_REQUESTS, primaryNRFRetryCount=0,
nonPrimaryNRFRetryCount=0, alternateNRFRetryCount=-1,
errorReasonsForFailure=[503, 504, 500, SocketTimeoutException,
JsonProcessingException, UnknownHostException, NoRouteToHostException],
gatewayErrorCodes=[503],
requestTimeout=10]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.
Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T
06:07:26.027+0000"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":27597244},"thread":"pool-10-
thread-1","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientA
pi","message":"Sending request to NRF nf1stub.gi-dina.svc:8080, routeCount=0,
attempt=0","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogg
er","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:07:26.
027+0000"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":27630106},"thread":"pool-10-
thread-1","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientA
pi","message":"trigger","endOfBatch":false,"loggerFqcn":"org.apache.logging.sl
f4j.Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12
-13T06:07:26.027+0000"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":27663824},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Returning retryConfig for ALL_REQUESTS NRFRetryConfig
[serviceRequestType=ALL_REQUESTS, primaryNRFRetryCount=0,
nonPrimaryNRFRetryCount=0, alternateNRFRetryCount=-1,
errorReasonsForFailure=[503, 504, 500, SocketTimeoutException,
JsonProcessingException, UnknownHostException, NoRouteToHostException],
gatewayErrorCodes=[503],
requestTimeout=10]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.
Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T
06:07:26.027+0000"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":60172668},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Request returned response with status code :
200","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","t
hreadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:07:26.060+00
00"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":65762172},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Successful Response code
received","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogge
r","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:07:26.0
65+0000"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":65848611},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"sendOnDemandNfDiscoverRequest with searchData:
{\"targetNfType\":\"AMF\",\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":

```
false,\"serviceInfoParamsPresent\":false,\"enableF3\":true,\"enableF5\":false,
\"targetNfSetId\":\"set001.region48.amfset.5gc.mnc012.mcc345\",\"retentionPeri
od\":5000,\"rawQueryParameters\":\"target-nf-type=AMF&requester-nf-
type=PCF&target-nf-set-
id=set001.region48.amfset.5gc.mnc012.mcc345\",\"forceRediscoveryEnabled\":fals
e} returned status code :
200","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","t
hreadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:07:26.065+00
00"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":66116142},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.nrf
.GetNrfDiscoveryResponseUseCase","message":"Received Discovery Result. Code:
200, Body (SearchResult): {\"validityPeriod\": 1800, \"nfInstances\":
[{\"nfInstanceId\": \"13515195-c537-4645-9b97-96ec797f111a\",
\"nfSetIdList\": [\"set001.region48.amfset.5gc.mnc012.mcc345\"],
\"allowedNfTypes\": [\"PCF\", \"SMF\"], \"nfType\": \"AMF\", \"plmnList\":
[{\"mcc\": \"450\", \"mnc\": \"05\"}], \"nfStatus\": \"REGISTERED\",
\"fqdn\": \"nf1stub.gi-dina.svc\", \"priority\": 1, \"capacity\": 100,
\"load\": 80, \"locality\": \"2\", \"amfInfo\": {\"amfRegionId\": \"01\",
\"amfSetId\": \"001\", \"guamiList\": [{\"plmnId\": {\"mcc\": \"450\",
\"mnc\": \"05\"}, \"amfId\": \"010041\"}]}, \"nfServices\":
[{\"serviceInstanceId\": \"aaaa-bbbb-cccc-dddd\", \"serviceName\": \"namf-
comm\", \"versions\": [{\"apiVersionInUri\": \"v1\", \"apiFullVersion\":
\"1.1.0\", \"expiry\": \"2020-07-30T12:09:55.65Z\"}], \"scheme\": \"http\",
\"nfServiceStatus\": \"REGISTERED\", \"ipEndPoints\": [{\"ipv4Address\":
null, \"ipv6Address\": null, \"transport\": \"TCP\", \"port\": 8080}],
\"allowedPlmns\": [{\"mcc\": \"450\", \"mnc\": \"05\"}], \"priority\": 2,
\"capacity\": 100, \"load\": 80, \"locality\": \"2\"}]}, {\"nfInstanceId\":
\"13515195-c537-4645-9b97-96ec797f1111\", \"nfSetIdList\":
[\"set001.region48.amfset.5gc.mnc012.mcc345\"], \"allowedNfTypes\": [\"PCF\",
\"SMF\"], \"nfType\": \"AMF\", \"plmnList\": [{\"mcc\": \"450\", \"mnc\":
\"05\"}], \"nfStatus\": \"REGISTERED\", \"fqdn\": \"nf12stub.gi-dina.svc\",
\"priority\": 2, \"capacity\": 100, \"load\": 80, \"locality\": \"2\",
\"amfInfo\": {\"amfRegionId\": \"01\", \"amfSetId\": \"001\", \"guamiList\":
[{\"plmnId\": {\"mcc\": \"450\", \"mnc\": \"05\"}, \"amfId\": \"010041\"}]},
\"nfServices\": [{\"serviceInstanceId\": \"aaaa-bbbb-cccc-dddd\",
\"serviceName\": \"namf-comm\", \"versions\": [{\"apiVersionInUri\": \"v1\",
\"apiFullVersion\": \"1.1.0\", \"expiry\": \"2020-07-30T12:09:55.65Z\"}],
\"scheme\": \"http\", \"nfServiceStatus\": \"REGISTERED\", \"ipEndPoints\":
[{\"ipv4Address\": null, \"ipv6Address\": null, \"transport\": \"TCP\",
\"port\": 8080}], \"allowedPlmns\": [{\"mcc\": \"450\", \"mnc\": \"05\"}],
\"priority\": 2, \"capacity\": 100, \"load\": 80, \"locality\": \"2\"}]},
{\"nfInstanceId\": \"13515195-c537-4645-9b97-96ec797f2222\", \"nfSetIdList\":
[\"set001.region48.amfset.5gc.mnc012.mcc345\"], \"allowedNfTypes\": [\"PCF\",
\"SMF\"], \"nfType\": \"AMF\", \"plmnList\": [{\"mcc\": \"450\", \"mnc\":
\"05\"}], \"nfStatus\": \"REGISTERED\", \"fqdn\": \"nf2stub.gi-dina.svc\",
\"priority\": 1, \"capacity\": 100, \"load\": 80, \"locality\": \"1\",
\"amfInfo\": {\"amfRegionId\": \"01\", \"amfSetId\": \"001\", \"guamiList\":
[{\"plmnId\": {\"mcc\": \"450\", \"mnc\": \"05\"}, \"amfId\": \"010041\"}]},
\"nfServices\": [{\"serviceInstanceId\": \"aaaa-bbbb-cccc-dddd\",
\"serviceName\": \"namf-comm\", \"versions\": [{\"apiVersionInUri\": \"v1\",
\"apiFullVersion\": \"1.1.0\", \"expiry\": \"2020-07-30T12:09:55.65Z\"}],
\"scheme\": \"http\", \"nfServiceStatus\": \"REGISTERED\", \"ipEndPoints\":
[{\"ipv4Address\": null, \"ipv6Address\": null, \"transport\": \"TCP\",
\"port\": 8080}], \"allowedPlmns\": [{\"mcc\": \"450\", \"mnc\": \"05\"}],
```

\"priority\": 1, \"capacity\": 100, \"load\": 80, \"locality\": \"1\"}]},
{\"nfInstanceId\": \"13515195-c537-4645-9b97-96ec797f3333\", \"nfSetIdList\":
[\"set001.region48.amfset.5gc.mnc012.mcc345\"], \"allowedNfTypes\": [\"PCF\",
\"SMF\"], \"nfType\": \"AMF\", \"plmnList\": [{\"mcc\": \"450\", \"mnc\":
\"05\"}], \"nfStatus\": \"REGISTERED\", \"fqdn\": \"nf22stub.gi-dina.svc\",
\"priority\": 2, \"capacity\": 100, \"load\": 80, \"locality\": \"1\",
\"amfInfo\": {\"amfRegionId\": \"01\", \"amfSetId\": \"001\", \"guamiList\":
[{\"plmnId\": {\"mcc\": \"450\", \"mnc\": \"05\"}, \"amfId\": \"010041\"}]},
\"nfServices\": [{\"serviceInstanceId\": \"aaaa-bbbb-cccc-dddd\",
\"serviceName\": \"namf-comm\", \"versions\": [{\"apiVersionInUri\": \"v1\",
\"apiFullVersion\": \"1.1.0\", \"expiry\": \"2020-07-30T12:09:55.65Z\"}],
\"scheme\": \"http\", \"nfServiceStatus\": \"REGISTERED\", \"ipEndPoints\":
[{\"ipv4Address\": null, \"ipv6Address\": null, \"transport\": \"TCP\",
\"port\": 8080}], \"allowedPlmns\": [{\"mcc\": \"450\", \"mnc\": \"05\"}],
\"priority\": 2, \"capacity\": 100, \"load\": 80, \"locality\": \"1\"}]},
{\"nfInstanceId\": \"13515195-c537-4645-9b97-96ec797f4444\", \"nfSetIdList\":
[\"set001.region48.amfset.5gc.mnc012.mcc345\"], \"allowedNfTypes\": [\"PCF\",
\"SMF\"], \"nfType\": \"AMF\", \"plmnList\": [{\"mcc\": \"450\", \"mnc\":
\"05\"}], \"nfStatus\": \"REGISTERED\", \"fqdn\": \"nf21stub.gi-dina.svc\",
\"priority\": 5, \"capacity\": 100, \"load\": 80, \"locality\": \"1\",
\"amfInfo\": {\"amfRegionId\": \"01\", \"amfSetId\": \"001\", \"guamiList\":
[{\"plmnId\": {\"mcc\": \"450\", \"mnc\": \"05\"}, \"amfId\": \"010041\"}]},
\"nfServices\": [{\"serviceInstanceId\": \"aaaa-bbbb-cccc-dddd\",
\"serviceName\": \"namf-comm\", \"versions\": [{\"apiVersionInUri\": \"v1\",
\"apiFullVersion\": \"1.1.0\", \"expiry\": \"2020-07-30T12:09:55.65Z\"}],
\"scheme\": \"http\", \"nfServiceStatus\": \"REGISTERED\", \"ipEndPoints\":
[{\"ipv4Address\": null, \"ipv6Address\": null, \"transport\": \"TCP\",
\"port\": 8080}], \"allowedPlmns\": [{\"mcc\": \"450\", \"mnc\": \"05\"}],
\"priority\": 5, \"capacity\": 100, \"load\": 80, \"locality\":
\"1\"}]}]}","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLog
ger","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:07:26.
066+0000"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":67989020},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.nrf
.GetNrfDiscoveryResponseUseCase","message":"Dynamic discovery cache enabled :
false","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger",
"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:07:26.067+0
000"}
{"instant":
{"epochSecond":1702447646,"nanoOfSecond":124766789},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Discovery Response fetched :
DiscoverySearchResult{queryParams='target-nf-type=AMF&requester-nf-
type=PCF&target-nf-set-id=set001.region48.amfset.5gc.mnc012.mcc345',
searchResult={\"validityPeriod\":1800,\"nfInstances\":
[{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797f111a\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf1stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":1,\"capacity\":100,\"load\":80,\"locality\":\"2
\",\"nfSetIdList\":[\"set001.region48.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":

[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"priority\":2,\"capacity\":100,\"load\":80
,\"locality\":\"2\"}]},{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797f1111\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf12stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":2,\"capacity\":100,\"load\":80,\"locality\":\"2
\",\"nfSetIdList\":[\"set001.region48.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"priority\":2,\"capacity\":100,\"load\":80
,\"locality\":\"2\"}]},{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797f2222\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf2stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":1,\"capacity\":100,\"load\":80,\"locality\":\"1
\",\"nfSetIdList\":[\"set001.region48.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"priority\":1,\"capacity\":100,\"load\":80
,\"locality\":\"1\"}]},{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797f3333\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf22stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":2,\"capacity\":100,\"load\":80,\"locality\":\"1
\",\"nfSetIdList\":[\"set001.region48.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"priority\":2,\"capacity\":100,\"load\":80
,\"locality\":\"1\"}]},{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797f4444\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf21stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":5,\"capacity\":100,\"load\":80,\"locality\":\"1
\",\"nfSetIdList\":[\"set001.region48.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":

[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"priority\":5,\"capacity\":100,\"load\":80
,\"locality\":\"1\"}]}],\"nrfSupportedFeatures\":\"72\"},
sourceType=NRF}","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log
4jLogger","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:0
7:26.124+0000"}

Discovery of AMF for UE policy association for the second time:

{"instant":
{"epochSecond":1702447706,"nanoOfSecond":365554062},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryRule","message":"Checking if cache applies for a discovery
request (DiscoveryCache feature enabled: true, OC-Force-Rediscovery
requested:
false)","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger"
,"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:08:26.365+
0000"}
{"instant":
{"epochSecond":1702447706,"nanoOfSecond":365614917},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryRule","message":"Query Parameters validation. IsFullyMatch ->
true, Params from request -> [target-nf-type, requester-nf-type, target-nf-
set-id], Params from config -> [target-nf-type, requester-nf-type, target-nf-
set-
id]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","t
hreadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:08:26.365+000
0"}
{"instant":
{"epochSecond":1702447706,"nanoOfSecond":365677878},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryrul
es.DiscoveryRule","message":"DiscoveryCache is enabled and queryParameters
matched. Proceeding with DiscoveryResponse retrieval from
memory","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger"
,"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:08:26.365+
0000"}
{"instant":
{"epochSecond":1702447706,"nanoOfSecond":365705820},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryEvaluator","message":"Rule applied:
CacheRule","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogg
er","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:08:26.3
65+0000"}
{"instant":
{"epochSecond":1702447706,"nanoOfSecond":365727980},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.CacheRule","message":"Applying Cache Rule, grabbing NRF response from
Cache","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger",
"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:08:26.365+0
000"}
{"instant":
{"epochSecond":1702447706,"nanoOfSecond":370207433},"thread":"XNIO-1

task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Successful retrieval of
Discovery Response stored in Cache/DB, proceeding with
validation","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLog
ger","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:08:26.
370+0000"}
{"instant":
{"epochSecond":1702447706,"nanoOfSecond":370252164},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Discovery Response retrieved
from Cache/DB isAboutToExpire value : false
","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","thre
adId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:08:26.370+0000"}
{"instant":
{"epochSecond":1702447706,"nanoOfSecond":370285114},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Discovery Response retrieved
from Cache/DB is expired value : false
","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","thre
adId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:08:26.370+0000"}
{"instant":
{"epochSecond":1702447706,"nanoOfSecond":378912228},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Discovery Response fetched :
DiscoverySearchResult{queryParams='target-nf-type=AMF&requester-nf-
type=PCF&target-nf-set-id=set001.region48.amfset.5gc.mnc012.mcc345',
searchResult={\"validityPeriod\":1800,\"nfInstances\":
[{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797f111a\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf1stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":1,\"capacity\":100,\"load\":80,\"locality\":\"2
\",\"nfSetIdList\":[\"set001.region48.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"priority\":2,\"capacity\":100,\"load\":80
,\"locality\":\"2\"}]},{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797f1111\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf12stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":2,\"capacity\":100,\"load\":80,\"locality\":\"2
\",\"nfSetIdList\":[\"set001.region48.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"priority\":2,\"capacity\":100,\"load\":80
,\"locality\":\"2\"}]},{\"nfInstanceId\":\"13515195-

c537-4645-9b97-96ec797f2222\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf2stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":1,\"capacity\":100,\"load\":80,\"locality\":\"1
\",\"nfSetIdList\":[\"set001.region48.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"priority\":1,\"capacity\":100,\"load\":80
,\"locality\":\"1\"}]},{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797f3333\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf22stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":2,\"capacity\":100,\"load\":80,\"locality\":\"1
\",\"nfSetIdList\":[\"set001.region48.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"priority\":2,\"capacity\":100,\"load\":80
,\"locality\":\"1\"}]},{\"nfInstanceId\":\"13515195-
c537-4645-9b97-96ec797f4444\",\"nfType\":\"AMF\",\"nfStatus\":\"REGISTERED\",\
"plmnList\":[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"fqdn\":\"nf21stub.gi-
dina.svc\",\"allowedNfTypes\":
[\"PCF\",\"SMF\"],\"priority\":5,\"capacity\":100,\"load\":80,\"locality\":\"1
\",\"nfSetIdList\":[\"set001.region48.amfset.5gc.mnc012.mcc345\"],\"amfInfo\":
{\"amfSetId\":\"001\",\"amfRegionId\":\"01\",\"guamiList\":[{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"}]},\"nfServices\":
[{\"serviceInstanceId\":\"aaaa-bbbb-cccc-dddd\",\"serviceName\":\"namf-
comm\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.1.0\",\"expiry\":\"2020-07
-30T12:09:55.650Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\",\
"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedPlmns\":
[{\"mcc\":\"450\",\"mnc\":\"05\"}],\"priority\":5,\"capacity\":100,\"load\":80
,\"locality\":\"1\"}]}],\"nrfSupportedFeatures\":\"72\"},
sourceType=CACHE}","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.L
og4jLogger","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06
:08:26.378+0000"}

UDR:

Discovery of UDR for th first time:

```
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":703559369},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryRule","message":"Checking if cache applies for a discovery
request (DiscoveryCache feature enabled: true, OC-Force-Rediscovery
requested:
false)","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger"
,"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:11:51.703+
0000"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":703624658},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryRule","message":"Query Parameters validation. IsFullyMatch ->
true, Params from request -> [target-nf-type, requester-nf-type, service-
names, target-nf-set-id], Params from config -> [target-nf-type, requester-nf-
type, service-names, target-nf-set-
id]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","t
hreadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:11:51.703+000
0"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":703677942},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryrul
es.DiscoveryRule","message":"DiscoveryCache is enabled and queryParameters
matched. Proceeding with DiscoveryResponse retrieval from
memory","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger"
,"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:11:51.703+
0000"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":703715314},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryEvaluator","message":"Rule applied:
CacheRule","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogg
er","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:11:51.7
03+0000"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":703735621},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.CacheRule","message":"Applying Cache Rule, grabbing NRF response from
Cache","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger",
"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:11:51.703+0
000"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":711068647},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Unsuccessful retrieval of
Discovery Response from Cache/DB, proceeding retrieval from NRF and saving
response in Cache/
DB","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","th
readId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:11:51.711+0000
"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":711135709},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.nrf
.GetNrfDiscoveryResponseUseCase","message":"SearchData :
{\"targetNfType\":\"UDR\",\"serviceNames\":[\"nudr-
```

dr\"],\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":false,\"serviceInfoP
aramsPresent\":true,\"enableF3\":true,\"enableF5\":false,\"targetNfSetId\":\"s
et001.udrset.5gc.mnc012.mcc345\",\"retentionPeriod\":5000,\"rawQueryParameters
\":\"target-nf-type=UDR&requester-nf-type=PCF&service-names=nudr-dr&target-nf-
set-
id=set001.udrset.5gc.mnc012.mcc345\",\"forceRediscoveryEnabled\":false}","endO
fBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":9
0,"threadPriority":5,"messageTimestamp":"2023-12-13T06:11:51.711+0000"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":711380511},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Entering sendOnDemandNfDiscoverRequest
function","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogge
r","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:11:51.7
11+0000"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":712130386},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Send NfDiscover request with searchData:
{\"targetNfType\":\"UDR\",\"serviceNames\":[\"nudr-
dr\"],\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":false,\"serviceInfoP
aramsPresent\":true,\"enableF3\":true,\"enableF5\":false,\"targetNfSetId\":\"s
et001.udrset.5gc.mnc012.mcc345\",\"retentionPeriod\":5000,\"rawQueryParameters
\":\"target-nf-type=UDR&requester-nf-type=PCF&service-names=nudr-dr&target-nf-
set-
id=set001.udrset.5gc.mnc012.mcc345\",\"forceRediscoveryEnabled\":false}","endO
fBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":1
01,"threadPriority":5,"messageTimestamp":"2023-12-13T06:11:51.712+0000"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":712257461},"thread":"pool-10-
thread-1","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientA
pi","message":"sendRequestToNrf nrfRequest : NRFRequest [scheme=http,
rootURI=nf1stub.gi-dina.svc:8080, resourcePath=/nnrf-disc/v1/nf-instances,
method=GET, body=null, headers=[],
searchData={\"targetNfType\":\"UDR\",\"serviceNames\":[\"nudr-
dr\"],\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":false,\"serviceInfoP
aramsPresent\":true,\"enableF3\":true,\"enableF5\":false,\"targetNfSetId\":\"s
et001.udrset.5gc.mnc012.mcc345\",\"retentionPeriod\":5000,\"rawQueryParameters
\":\"target-nf-type=UDR&requester-nf-type=PCF&service-names=nudr-dr&target-nf-
set-id=set001.udrset.5gc.mnc012.mcc345\",\"forceRediscoveryEnabled\":false},
routeCount=0,
requestType=NFDISCOVER]","endOfBatch":false,"loggerFqcn":"org.apache.logging.s
lf4j.Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-1
2-13T06:11:51.712+0000"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":712395565},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Returning retryConfig for ALL_REQUESTS NRFRetryConfig
[serviceRequestType=ALL_REQUESTS, primaryNRFRetryCount=0,
nonPrimaryNRFRetryCount=0, alternateNRFRetryCount=-1,
errorReasonsForFailure=[503, 504, 500, SocketTimeoutException,
JsonProcessingException, UnknownHostException, NoRouteToHostException],
gatewayErrorCodes=[503],
requestTimeout=10]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.
Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T
06:11:51.712+0000"}

{"instant":
{"epochSecond":1702447911,"nanoOfSecond":712450944},"thread":"pool-10-
thread-1","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientA
pi","message":"Sending request to NRF nf1stub.gi-dina.svc:8080, routeCount=0,
attempt=0","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogg
er","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:11:51.
712+0000"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":712478555},"thread":"pool-10-
thread-1","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientA
pi","message":"trigger","endOfBatch":false,"loggerFqcn":"org.apache.logging.sl
f4j.Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12
-13T06:11:51.712+0000"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":712505815},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Returning retryConfig for ALL_REQUESTS NRFRetryConfig
[serviceRequestType=ALL_REQUESTS, primaryNRFRetryCount=0,
nonPrimaryNRFRetryCount=0, alternateNRFRetryCount=-1,
errorReasonsForFailure=[503, 504, 500, SocketTimeoutException,
JsonProcessingException, UnknownHostException, NoRouteToHostException],
gatewayErrorCodes=[503],
requestTimeout=10]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.
Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T
06:11:51.712+0000"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":731529927},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Request returned response with status code :
200","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","t
hreadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:11:51.731+00
00"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":736236543},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Successful Response code
received","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogge
r","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:11:51.7
36+0000"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":736285120},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"sendOnDemandNfDiscoverRequest with searchData:
{\"targetNfType\":\"UDR\",\"serviceNames\":[\"nudr-
dr\"],\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":false,\"serviceInfoP
aramsPresent\":true,\"enableF3\":true,\"enableF5\":false,\"targetNfSetId\":\"s
et001.udrset.5gc.mnc012.mcc345\",\"retentionPeriod\":5000,\"rawQueryParameters
\":\"target-nf-type=UDR&requester-nf-type=PCF&service-names=nudr-dr&target-nf-
set-id=set001.udrset.5gc.mnc012.mcc345\",\"forceRediscoveryEnabled\":false}
returned status code :
200","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","t
hreadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:11:51.736+00
00"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":736499801},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.nrf

.GetNrfDiscoveryResponseUseCase","message":"Received Discovery Result. Code: 200, Body (SearchResult): {\"validityPeriod\": 100, \"nfInstances\": [{\"nfInstanceId\": \"fe7d992b-0541-4c7d-ab84-555550000000\", \"nfSetIdList\": [\"set001.udrset.5gc.mnc012.mcc345\"], \"nfType\": \"UDR\", \"nfStatus\": \"REGISTERED\", \"plmnList\": null, \"nsiList\": null, \"fqdn\": \"nf2stub.gi-dina.svc\", \"interPlmnFqdn\": null, \"ipv4Addresses\": null, \"ipv6Addresses\": null, \"priority\": 1, \"capacity\": 50, \"load\": 10, \"locality\": \"1\", \"pcfInfo\": null, \"udmInfo\": null, \"ausfInfo\": null, \"amfInfo\": null, \"smfInfo\": null, \"upfInfo\": null, \"groupId\": \"udr1\", \"udrInfo\": {\"supiRanges\": [{\"start\": \"450081000000000\", \"end\": \"450081002000000\"}], \"gpsiRanges\": [{\"start\": \"13100000000\", \"end\": \"13102000000\"}], \"supportedDataSets\": [\"POLICY\"]}, \"bsfInfo\": null, \"customInfo\": null, \"recoveryTime\": null, \"nfServices\": [{\"serviceInstanceId\": \"94d8d19a-b0d9-4d6d-994f-39a49ed5c111\", \"serviceName\": \"nudr-dr\", \"versions\": [{\"apiVersionInUri\": \"v1\", \"apiFullVersion\": \"1.15.1.0\", \"expiry\": \"2019-08-03T18:55:08.871+0000\"}], \"scheme\": \"http\", \"nfServiceStatus\": \"REGISTERED\", \"fqdn\": null, \"interPlmnFqdn\": null, \"ipEndPoints\": [{\"ipv4Address\": null, \"ipv6Address\": null, \"transport\": \"TCP\", \"port\": 8080}], \"apiPrefix\": null, \"defaultNotificationSubscriptions\": null, \"allowedPlmns\": null, \"allowedNfTypes\": [\"CHF\", \"PCF\"], \"allowedNfDomains\": null, \"allowedNssais\": null, \"priority\": 1, \"capacity\": 100, \"load\": 50, \"locality\": \"1\", \"recoveryTime\": 1542876663222, \"supportedFeatures\": null}]}, {\"nfInstanceId\": \"fe7d992b-0541-4c7d-ab84-555551111111\", \"nfSetIdList\": [\"set001.udrset.5gc.mnc012.mcc345\"], \"nfType\": \"UDR\", \"nfStatus\": \"REGISTERED\", \"plmnList\": null, \"nsiList\": null, \"fqdn\": \"nf21stub.gi-dina.svc\", \"interPlmnFqdn\": null, \"ipv4Addresses\": null, \"ipv6Addresses\": null, \"priority\": 2, \"capacity\": 50, \"load\": 20, \"locality\": \"2\", \"pcfInfo\": null, \"udmInfo\": null, \"ausfInfo\": null, \"amfInfo\": null, \"smfInfo\": null, \"upfInfo\": null, \"groupId\": \"udr1\", \"udrInfo\": {\"supiRanges\": [{\"start\": \"450081000000000\", \"end\": \"450081002000000\"}], \"gpsiRanges\": [{\"start\": \"13100000000\", \"end\": \"13102000000\"}], \"supportedDataSets\": [\"POLICY\"]}, \"bsfInfo\": null, \"customInfo\": null, \"recoveryTime\": null, \"nfServices\": [{\"serviceInstanceId\": \"94d8d19a-b0d9-4d6d-994f-39a49ed5c111\", \"serviceName\": \"nudr-dr\", \"versions\": [{\"apiVersionInUri\": \"v1\", \"apiFullVersion\": \"1.15.1.0\", \"expiry\": \"2019-08-03T18:55:08.871+0000\"}], \"scheme\": \"http\", \"nfServiceStatus\": \"REGISTERED\", \"fqdn\": null, \"interPlmnFqdn\": null, \"ipEndPoints\": [{\"ipv4Address\": null, \"ipv6Address\": null, \"transport\": \"TCP\", \"port\": 8080}], \"apiPrefix\": null, \"defaultNotificationSubscriptions\": null, \"allowedPlmns\": null, \"allowedNfTypes\": [\"CHF\", \"PCF\"], \"allowedNfDomains\": null, \"allowedNssais\": null, \"priority\": 2, \"capacity\": 50, \"load\": 20, \"recoveryTime\": 1542876663222, \"supportedFeatures\": null}]}, {\"nfInstanceId\": \"fe7d992b-0541-4c7d-ab84-555553333333\", \"nfSetIdList\": [\"set001.udrset.5gc.mnc012.mcc345\"], \"nfType\": \"UDR\", \"nfStatus\": \"REGISTERED\", \"plmnList\": null, \"nsiList\": null, \"fqdn\": \"nf11stub.gi-dina.svc\", \"interPlmnFqdn\": null, \"ipv4Addresses\": null, \"ipv6Addresses\": null, \"priority\": 4, \"capacity\": 50, \"load\": 10, \"locality\": \"1\", \"pcfInfo\": null, \"udmInfo\": null, \"ausfInfo\": null, \"amfInfo\": null, \"smfInfo\": null, \"upfInfo\": null, \"groupId\": \"udr1\", \"udrInfo\": {\"supiRanges\": [{\"start\": \"450081000000000\", \"end\": \"450081002000000\"}], \"gpsiRanges\": [{\"start\": \"13100000000\",

\"end\": \"13102000000\"}], \"supportedDataSets\": [\"POLICY\"]},
\"bsfInfo\": null, \"customInfo\": null, \"recoveryTime\": null,
\"nfServices\": [{\"serviceInstanceId\": \"94d8d19a-
b0d9-4d6d-994f-39a49ed5c111\", \"serviceName\": \"nudr-dr\", \"versions\":
[{\"apiVersionInUri\": \"v1\", \"apiFullVersion\": \"1.15.1.0\", \"expiry\":
\"2019-08-03T18:55:08.871+0000\"}], \"scheme\": \"http\",
\"nfServiceStatus\": \"REGISTERED\", \"fqdn\": null, \"interPlmnFqdn\": null,
\"ipEndPoints\": [{\"ipv4Address\": null, \"ipv6Address\": null,
\"transport\": \"TCP\", \"port\": 8080}], \"apiPrefix\": null,
\"defaultNotificationSubscriptions\": null, \"allowedPlmns\": null,
\"allowedNfTypes\": [\"CHF\", \"PCF\"], \"allowedNfDomains\": null,
\"allowedNssais\": null, \"priority\": 4, \"capacity\": 50, \"load\": 10,
\"recoveryTime\": 1542876663222, \"supportedFeatures\": null}]},
{\"nfInstanceId\": \"fe7d992b-0541-4c7d-ab84-555554444444\", \"nfSetIdList\":
[\"set002.udrset.5gc.mnc012.mcc345\"], \"nfType\": \"UDR\", \"nfStatus\":
\"REGISTERED\", \"plmnList\": null, \"nsiList\": null, \"fqdn\":
\"nf12stub.gi-dina.svc\", \"interPlmnFqdn\": null, \"ipv4Addresses\": null,
\"ipv6Addresses\": null, \"priority\": 5, \"capacity\": 50, \"load\": 40,
\"locality\": \"2\", \"pcfInfo\": null, \"udmInfo\": null, \"ausfInfo\":
null, \"amfInfo\": null, \"smfInfo\": null, \"upfInfo\": null, \"groupId\":
\"udr1\", \"udrInfo\": {\"supiRanges\": [{\"start\": \"450081000000000\",
\"end\": \"450081002000000\"}], \"gpsiRanges\": [{\"start\": \"13100000000\",
\"end\": \"13102000000\"}], \"supportedDataSets\": [\"POLICY\"]},
\"bsfInfo\": null, \"customInfo\": null, \"recoveryTime\": null,
\"nfServices\": [{\"serviceInstanceId\": \"94d8d19a-
b0d9-4d6d-994f-39a49ed5c111\", \"serviceName\": \"nudr-dr\", \"versions\":
[{\"apiVersionInUri\": \"v1\", \"apiFullVersion\": \"1.15.1.0\", \"expiry\":
\"2019-08-03T18:55:08.871+0000\"}], \"scheme\": \"http\",
\"nfServiceStatus\": \"REGISTERED\", \"fqdn\": null, \"interPlmnFqdn\": null,
\"ipEndPoints\": [{\"ipv4Address\": null, \"ipv6Address\": null,
\"transport\": \"TCP\", \"port\": 8080}], \"apiPrefix\": null,
\"defaultNotificationSubscriptions\": null, \"allowedPlmns\": null,
\"allowedNfTypes\": [\"CHF\", \"PCF\"], \"allowedNfDomains\": null,
\"allowedNssais\": null, \"priority\": 5, \"capacity\": 50, \"load\": 40,
\"recoveryTime\": 1542876663222, \"supportedFeatures\": null}]},
{\"nfInstanceId\": \"fe7d992b-0541-4c7d-ab84-555552222222\", \"nfSetIdList\":
[\"set002.udrset.5gc.mnc012.mcc345\"], \"nfType\": \"UDR\", \"nfStatus\":
\"REGISTERED\", \"plmnList\": null, \"nsiList\": null, \"fqdn\":
\"nf22stub.gi-dina.svc\", \"interPlmnFqdn\": null, \"ipv4Addresses\": null,
\"ipv6Addresses\": null, \"priority\": 3, \"capacity\": 50, \"load\": 30,
\"locality\": \"1\", \"pcfInfo\": null, \"udmInfo\": null, \"ausfInfo\":
null, \"amfInfo\": null, \"smfInfo\": null, \"upfInfo\": null, \"groupId\":
\"udr1\", \"udrInfo\": {\"supiRanges\": [{\"start\": \"450081000000000\",
\"end\": \"450081002000000\"}], \"gpsiRanges\": [{\"start\": \"13100000000\",
\"end\": \"13102000000\"}], \"supportedDataSets\": [\"POLICY\"]},
\"bsfInfo\": null, \"customInfo\": null, \"recoveryTime\": null,
\"nfServices\": [{\"serviceInstanceId\": \"94d8d19a-
b0d9-4d6d-994f-39a49ed5c111\", \"serviceName\": \"nudr-dr\", \"versions\":
[{\"apiVersionInUri\": \"v1\", \"apiFullVersion\": \"1.15.1.0\", \"expiry\":
\"2019-08-03T18:55:08.871+0000\"}], \"scheme\": \"http\",
\"nfServiceStatus\": \"REGISTERED\", \"fqdn\": null, \"interPlmnFqdn\": null,
\"ipEndPoints\": [{\"ipv4Address\": null, \"ipv6Address\": null,
\"transport\": \"TCP\", \"port\": 8080}], \"apiPrefix\": null,
\"defaultNotificationSubscriptions\": null, \"allowedPlmns\": null,
\"allowedNfTypes\": [\"CHF\", \"PCF\"], \"allowedNfDomains\": null,
\"allowedNssais\": null, \"priority\": 3, \"capacity\": 50, \"load\": 30,

\"recoveryTime\": 1542876663222, \"supportedFeatures\":
null}]}]}","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogg
er","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:11:51.7
36+0000"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":737221498},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.nrf
.GetNrfDiscoveryResponseUseCase","message":"Dynamic discovery cache enabled :
false","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger",
"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:11:51.737+0
000"}
{"instant":
{"epochSecond":1702447911,"nanoOfSecond":770303926},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Discovery Response fetched :
DiscoverySearchResult{queryParams='target-nf-type=UDR&requester-nf-
type=PCF&service-names=nudr-dr&target-nf-set-
id=set001.udrset.5gc.mnc012.mcc345',
searchResult={\"validityPeriod\":100,\"nfInstances\":
[{\"nfInstanceId\":\"fe7d992b-0541-4c7d-
ab84-555550000000\",\"nfType\":\"UDR\",\"nfStatus\":\"REGISTERED\",\"fqdn\":\"
nf2stub.gi-
dina.svc\",\"priority\":1,\"capacity\":50,\"load\":10,\"locality\":\"1\",\"nfS
etIdList\":[\"set001.udrset.5gc.mnc012.mcc345\"],\"udrInfo\":{\"supiRanges\":
[{\"start\":\"450081000000000\",\"end\":\"450081002000000\"}],\"gpsiRanges\":
[{\"start\":\"13100000000\",\"end\":\"13102000000\"}],\"supportedDataSets\":
[\"POLICY\"]},\"nfServices\":[{\"serviceInstanceId\":\"94d8d19a-
b0d9-4d6d-994f-39a49ed5c111\",\"serviceName\":\"nudr-dr\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.15.1.0\",\"expiry\":\"2019
-08-03T18:55:08.871Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\
",\"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedNfTypes\":
[\"CHF\",\"PCF\"],\"priority\":1,\"capacity\":100,\"load\":50,\"recoveryTime\"
:\"2018-11-22T08:51:03.222Z\",\"locality\":\"1\"}],\"groupId\":\"udr1\"},
{\"nfInstanceId\":\"fe7d992b-0541-4c7d-
ab84-555551111111\",\"nfType\":\"UDR\",\"nfStatus\":\"REGISTERED\",\"fqdn\":\"
nf21stub.gi-
dina.svc\",\"priority\":2,\"capacity\":50,\"load\":20,\"locality\":\"2\",\"nfS
etIdList\":[\"set001.udrset.5gc.mnc012.mcc345\"],\"udrInfo\":{\"supiRanges\":
[{\"start\":\"450081000000000\",\"end\":\"450081002000000\"}],\"gpsiRanges\":
[{\"start\":\"13100000000\",\"end\":\"13102000000\"}],\"supportedDataSets\":
[\"POLICY\"]},\"nfServices\":[{\"serviceInstanceId\":\"94d8d19a-
b0d9-4d6d-994f-39a49ed5c111\",\"serviceName\":\"nudr-dr\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.15.1.0\",\"expiry\":\"2019
-08-03T18:55:08.871Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\
",\"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedNfTypes\":
[\"CHF\",\"PCF\"],\"priority\":2,\"capacity\":50,\"load\":20,\"recoveryTime\":
\"2018-11-22T08:51:03.222Z\"}],\"groupId\":\"udr1\"},
{\"nfInstanceId\":\"fe7d992b-0541-4c7d-
ab84-555553333333\",\"nfType\":\"UDR\",\"nfStatus\":\"REGISTERED\",\"fqdn\":\"
nf11stub.gi-
dina.svc\",\"priority\":4,\"capacity\":50,\"load\":10,\"locality\":\"1\",\"nfS
etIdList\":[\"set001.udrset.5gc.mnc012.mcc345\"],\"udrInfo\":{\"supiRanges\":
[{\"start\":\"450081000000000\",\"end\":\"450081002000000\"}],\"gpsiRanges\":
[{\"start\":\"13100000000\",\"end\":\"13102000000\"}],\"supportedDataSets\":
[\"POLICY\"]},\"nfServices\":[{\"serviceInstanceId\":\"94d8d19a-
b0d9-4d6d-994f-39a49ed5c111\",\"serviceName\":\"nudr-dr\",\"versions\":

[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.15.1.0\",\"expiry\":\"2019
-08-03T18:55:08.871Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\
",\"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedNfTypes\":
[\"CHF\",\"PCF\"],\"priority\":4,\"capacity\":50,\"load\":10,\"recoveryTime\":
\"2018-11-22T08:51:03.222Z\"}],\"groupId\":\"udr1\"},
{\"nfInstanceId\":\"fe7d992b-0541-4c7d-
ab84-555554444444\",\"nfType\":\"UDR\",\"nfStatus\":\"REGISTERED\",\"fqdn\":\"
nf12stub.gi-
dina.svc\",\"priority\":5,\"capacity\":50,\"load\":40,\"locality\":\"2\",\"nfS
etIdList\":[\"set002.udrset.5gc.mnc012.mcc345\"],\"udrInfo\":{\"supiRanges\":
[{\"start\":\"450081000000000\",\"end\":\"450081002000000\"}],\"gpsiRanges\":
[{\"start\":\"13100000000\",\"end\":\"13102000000\"}],\"supportedDataSets\":
[\"POLICY\"]},\"nfServices\":[{\"serviceInstanceId\":\"94d8d19a-
b0d9-4d6d-994f-39a49ed5c111\",\"serviceName\":\"nudr-dr\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.15.1.0\",\"expiry\":\"2019
-08-03T18:55:08.871Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\
",\"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedNfTypes\":
[\"CHF\",\"PCF\"],\"priority\":5,\"capacity\":50,\"load\":40,\"recoveryTime\":
\"2018-11-22T08:51:03.222Z\"}],\"groupId\":\"udr1\"},
{\"nfInstanceId\":\"fe7d992b-0541-4c7d-
ab84-555552222222\",\"nfType\":\"UDR\",\"nfStatus\":\"REGISTERED\",\"fqdn\":\"
nf22stub.gi-
dina.svc\",\"priority\":3,\"capacity\":50,\"load\":30,\"locality\":\"1\",\"nfS
etIdList\":[\"set002.udrset.5gc.mnc012.mcc345\"],\"udrInfo\":{\"supiRanges\":
[{\"start\":\"450081000000000\",\"end\":\"450081002000000\"}],\"gpsiRanges\":
[{\"start\":\"13100000000\",\"end\":\"13102000000\"}],\"supportedDataSets\":
[\"POLICY\"]},\"nfServices\":[{\"serviceInstanceId\":\"94d8d19a-
b0d9-4d6d-994f-39a49ed5c111\",\"serviceName\":\"nudr-dr\",\"versions\":
[{\"apiVersionInUri\":\"v1\",\"apiFullVersion\":\"1.15.1.0\",\"expiry\":\"2019
-08-03T18:55:08.871Z\"}],\"scheme\":\"http\",\"nfServiceStatus\":\"REGISTERED\
",\"ipEndPoints\":[{\"transport\":\"TCP\",\"port\":8080}],\"allowedNfTypes\":
[\"CHF\",\"PCF\"],\"priority\":3,\"capacity\":50,\"load\":30,\"recoveryTime\":
\"2018-11-22T08:51:03.222Z\"}],\"groupId\":\"udr1\"}],\"nrfSupportedFeatures\"
:\"72\"},
sourceType=NRF}","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log
4jLogger","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:1
1:51.770+0000"}

Discovery of UDR for the second time:

{"instant":
{"epochSecond":1702448069,"nanoOfSecond":173667948},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryRule","message":"Checking if cache applies for a discovery
request (DiscoveryCache feature enabled: true, OC-Force-Rediscovery
requested:
false)","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger"
,"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.173+
0000"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":173707825},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryRule","message":"Query Parameters validation. IsFullyMatch ->
true, Params from request -> [target-nf-type, requester-nf-type, service-
names, target-nf-set-id], Params from config -> [target-nf-type, requester-nf-

```
type, service-names, target-nf-set-
id]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","t
hreadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.173+000
0"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":173763918},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryrul
es.DiscoveryRule","message":"DiscoveryCache is enabled and queryParameters
matched. Proceeding with DiscoveryResponse retrieval from
memory","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger"
,"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.173+
0000"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":173787036},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.DiscoveryEvaluator","message":"Rule applied:
CacheRule","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogg
er","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.1
73+0000"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":173803555},"thread":"XNIO-1
task-2","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.core.discoveryru
les.CacheRule","message":"Applying Cache Rule, grabbing NRF response from
Cache","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger",
"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.173+0
000"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":178149415},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Successful retrieval of
Discovery Response stored in Cache/DB, proceeding with
validation","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLog
ger","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.
178+0000"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":178179123},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Discovery Response retrieved
from Cache/DB isAboutToExpire value : false
","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","thre
adId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.178+0000"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":178199339},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Discovery Response retrieved
from Cache/DB is expired value : true
","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","thre
adId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.178+0000"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":178405635},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.cac
he.GetCacheDiscoveryResponseUseCase","message":"Discovery Response has
expired, updating and retrieving from NRF to
Cache","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger",
"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.178+0
000"}
```

```
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":178450502},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.nrf
.GetNrfDiscoveryResponseUseCase","message":"SearchData :
{\"targetNfType\":\"UDR\",\"serviceNames\":[\"nudr-
dr\"],\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":false,\"serviceInfoP
aramsPresent\":true,\"enableF3\":true,\"enableF5\":false,\"targetNfSetId\":\"s
et001.udrset.5gc.mnc012.mcc345\",\"retentionPeriod\":5000,\"rawQueryParameters
\":\"target-nf-type=UDR&requester-nf-type=PCF&service-names=nudr-dr&target-nf-
set-
id=set001.udrset.5gc.mnc012.mcc345\",\"forceRediscoveryEnabled\":false}","endO
fBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":9
0,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.178+0000"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":178601367},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Entering sendOnDemandNfDiscoverRequest
function","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogge
r","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.1
78+0000"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":179243376},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Send NfDiscover request with searchData:
{\"targetNfType\":\"UDR\",\"serviceNames\":[\"nudr-
dr\"],\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":false,\"serviceInfoP
aramsPresent\":true,\"enableF3\":true,\"enableF5\":false,\"targetNfSetId\":\"s
et001.udrset.5gc.mnc012.mcc345\",\"retentionPeriod\":5000,\"rawQueryParameters
\":\"target-nf-type=UDR&requester-nf-type=PCF&service-names=nudr-dr&target-nf-
set-
id=set001.udrset.5gc.mnc012.mcc345\",\"forceRediscoveryEnabled\":false}","endO
fBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":1
01,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.179+0000"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":179374539},"thread":"pool-10-
thread-1","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientA
pi","message":"sendRequestToNrf nrfRequest : NRFRequest [scheme=http,
rootURI=nf1stub.gi-dina.svc:8080, resourcePath=/nnrf-disc/v1/nf-instances,
method=GET, body=null, headers=[],
searchData={\"targetNfType\":\"UDR\",\"serviceNames\":[\"nudr-
dr\"],\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":false,\"serviceInfoP
aramsPresent\":true,\"enableF3\":true,\"enableF5\":false,\"targetNfSetId\":\"s
et001.udrset.5gc.mnc012.mcc345\",\"retentionPeriod\":5000,\"rawQueryParameters
\":\"target-nf-type=UDR&requester-nf-type=PCF&service-names=nudr-dr&target-nf-
set-id=set001.udrset.5gc.mnc012.mcc345\",\"forceRediscoveryEnabled\":false},
routeCount=0,
requestType=NFDISCOVER]","endOfBatch":false,"loggerFqcn":"org.apache.logging.s
lf4j.Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-1
2-13T06:14:29.179+0000"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":179497176},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Returning retryConfig for ALL_REQUESTS NRFRetryConfig
[serviceRequestType=ALL_REQUESTS, primaryNRFRetryCount=0,
nonPrimaryNRFRetryCount=0, alternateNRFRetryCount=-1,
errorReasonsForFailure=[503, 504, 500, SocketTimeoutException,
```

JsonProcessingException, UnknownHostException, NoRouteToHostException],
gatewayErrorCodes=[503],
requestTimeout=10]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.
Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T
06:14:29.179+0000"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":179552046},"thread":"pool-10-
thread-1","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientA
pi","message":"Sending request to NRF nf1stub.gi-dina.svc:8080, routeCount=0,
attempt=0","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogg
er","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.
179+0000"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":179577169},"thread":"pool-10-
thread-1","level":"DEBUG","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientA
pi","message":"trigger","endOfBatch":false,"loggerFqcn":"org.apache.logging.sl
f4j.Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12
-13T06:14:29.179+0000"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":179601014},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Returning retryConfig for ALL_REQUESTS NRFRetryConfig
[serviceRequestType=ALL_REQUESTS, primaryNRFRetryCount=0,
nonPrimaryNRFRetryCount=0, alternateNRFRetryCount=-1,
errorReasonsForFailure=[503, 504, 500, SocketTimeoutException,
JsonProcessingException, UnknownHostException, NoRouteToHostException],
gatewayErrorCodes=[503],
requestTimeout=10]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.
Log4jLogger","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T
06:14:29.179+0000"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":197764440},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Request returned response with status code :
200","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","t
hreadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.197+00
00"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":203073867},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"Successful Response code
received","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogge
r","threadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.2
03+0000"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":203133658},"thread":"pool-10-
thread-1","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.api.NRFClientAp
i","message":"sendOnDemandNfDiscoverRequest with searchData:
{\"targetNfType\":\"UDR\",\"serviceNames\":[\"nudr-
dr\"],\"requesterNfType\":\"PCF\",\"nfInfoParamsPresent\":false,\"serviceInfoP
aramsPresent\":true,\"enableF3\":true,\"enableF5\":false,\"targetNfSetId\":\"s
et001.udrset.5gc.mnc012.mcc345\",\"retentionPeriod\":5000,\"rawQueryParameters
\":\"target-nf-type=UDR&requester-nf-type=PCF&service-names=nudr-dr&target-nf-
set-id=set001.udrset.5gc.mnc012.mcc345\",\"forceRediscoveryEnabled\":false}
returned status code :
200","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","t

hreadId":101,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.203+00
00"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":203337703},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.nrf
.GetNrfDiscoveryResponseUseCase","message":"Received Discovery Result. Code:
200, Body (SearchResult): {\"validityPeriod\": 100, \"nfInstances\":
[{\"nfInstanceId\": \"fe7d992b-0541-4c7d-ab84-555550000000\",
\"nfSetIdList\": [\"set001.udrset.5gc.mnc012.mcc345\"], \"nfType\": \"UDR\",
\"nfStatus\": \"REGISTERED\", \"plmnList\": null, \"nsiList\": null,
\"fqdn\": \"nf2stub.gi-dina.svc\", \"interPlmnFqdn\": null,
\"ipv4Addresses\": null, \"ipv6Addresses\": null, \"priority\": 1,
\"capacity\": 50, \"load\": 10, \"locality\": \"1\", \"pcfInfo\": null,
\"udmInfo\": null, \"ausfInfo\": null, \"amfInfo\": null, \"smfInfo\": null,
\"upfInfo\": null, \"groupId\": \"udr1\", \"udrInfo\": {\"supiRanges\":
[{\"start\": \"450081000000000\", \"end\": \"450081002000000\"}],
\"gpsiRanges\": [{\"start\": \"13100000000\", \"end\": \"13102000000\"}],
\"supportedDataSets\": [\"POLICY\"]}, \"bsfInfo\": null, \"customInfo\":
null, \"recoveryTime\": null, \"nfServices\": [{\"serviceInstanceId\":
\"94d8d19a-b0d9-4d6d-994f-39a49ed5c111\", \"serviceName\": \"nudr-dr\",
\"versions\": [{\"apiVersionInUri\": \"v1\", \"apiFullVersion\":
\"1.15.1.0\", \"expiry\": \"2019-08-03T18:55:08.871+0000\"}], \"scheme\":
\"http\", \"nfServiceStatus\": \"REGISTERED\", \"fqdn\": null,
\"interPlmnFqdn\": null, \"ipEndPoints\": [{\"ipv4Address\": null,
\"ipv6Address\": null, \"transport\": \"TCP\", \"port\": 8080}],
\"apiPrefix\": null, \"defaultNotificationSubscriptions\": null,
\"allowedPlmns\": null, \"allowedNfTypes\": [\"CHF\", \"PCF\"],
\"allowedNfDomains\": null, \"allowedNssais\": null, \"priority\": 1,
\"capacity\": 100, \"load\": 50, \"locality\": \"1\", \"recoveryTime\":
1542876663222, \"supportedFeatures\": null}]}, {\"nfInstanceId\":
\"fe7d992b-0541-4c7d-ab84-555551111111\", \"nfSetIdList\":
[\"set001.udrset.5gc.mnc012.mcc345\"], \"nfType\": \"UDR\", \"nfStatus\":
\"REGISTERED\", \"plmnList\": null, \"nsiList\": null, \"fqdn\":
\"nf21stub.gi-dina.svc\", \"interPlmnFqdn\": null, \"ipv4Addresses\": null,
\"ipv6Addresses\": null, \"priority\": 2, \"capacity\": 50, \"load\": 20,
\"locality\": \"2\", \"pcfInfo\": null, \"udmInfo\": null, \"ausfInfo\":
null, \"amfInfo\": null, \"smfInfo\": null, \"upfInfo\": null, \"groupId\":
\"udr1\", \"udrInfo\": {\"supiRanges\": [{\"start\": \"450081000000000\",
\"end\": \"450081002000000\"}], \"gpsiRanges\": [{\"start\": \"13100000000\",
\"end\": \"13102000000\"}], \"supportedDataSets\": [\"POLICY\"]},
\"bsfInfo\": null, \"customInfo\": null, \"recoveryTime\": null,
\"nfServices\": [{\"serviceInstanceId\": \"94d8d19a-
b0d9-4d6d-994f-39a49ed5c111\", \"serviceName\": \"nudr-dr\", \"versions\":
[{\"apiVersionInUri\": \"v1\", \"apiFullVersion\": \"1.15.1.0\", \"expiry\":
\"2019-08-03T18:55:08.871+0000\"}], \"scheme\": \"http\",
\"nfServiceStatus\": \"REGISTERED\", \"fqdn\": null, \"interPlmnFqdn\": null,
\"ipEndPoints\": [{\"ipv4Address\": null, \"ipv6Address\": null,
\"transport\": \"TCP\", \"port\": 8080}], \"apiPrefix\": null,
\"defaultNotificationSubscriptions\": null, \"allowedPlmns\": null,
\"allowedNfTypes\": [\"CHF\", \"PCF\"], \"allowedNfDomains\": null,
\"allowedNssais\": null, \"priority\": 2, \"capacity\": 50, \"load\": 20,
\"recoveryTime\": 1542876663222, \"supportedFeatures\": null}]},
{\"nfInstanceId\": \"fe7d992b-0541-4c7d-ab84-555553333333\", \"nfSetIdList\":
[\"set001.udrset.5gc.mnc012.mcc345\"], \"nfType\": \"UDR\", \"nfStatus\":
\"REGISTERED\", \"plmnList\": null, \"nsiList\": null, \"fqdn\":
\"nf11stub.gi-dina.svc\", \"interPlmnFqdn\": null, \"ipv4Addresses\": null,

\"ipv6Addresses\": null, \"priority\": 4, \"capacity\": 50, \"load\": 10,
\"locality\": \"1\", \"pcfInfo\": null, \"udmInfo\": null, \"ausfInfo\":
null, \"amfInfo\": null, \"smfInfo\": null, \"upfInfo\": null, \"groupId\":
\"udr1\", \"udrInfo\": {\"supiRanges\": [{\"start\": \"450081000000000\",
\"end\": \"450081002000000\"}], \"gpsiRanges\": [{\"start\": \"13100000000\",
\"end\": \"13102000000\"}], \"supportedDataSets\": [\"POLICY\"]},
\"bsfInfo\": null, \"customInfo\": null, \"recoveryTime\": null,
\"nfServices\": [{\"serviceInstanceId\": \"94d8d19a-
b0d9-4d6d-994f-39a49ed5c111\", \"serviceName\": \"nudr-dr\", \"versions\":
[{\"apiVersionInUri\": \"v1\", \"apiFullVersion\": \"1.15.1.0\", \"expiry\":
\"2019-08-03T18:55:08.871+0000\"}], \"scheme\": \"http\",
\"nfServiceStatus\": \"REGISTERED\", \"fqdn\": null, \"interPlmnFqdn\": null,
\"ipEndPoints\": [{\"ipv4Address\": null, \"ipv6Address\": null,
\"transport\": \"TCP\", \"port\": 8080}], \"apiPrefix\": null,
\"defaultNotificationSubscriptions\": null, \"allowedPlmns\": null,
\"allowedNfTypes\": [\"CHF\", \"PCF\"], \"allowedNfDomains\": null,
\"allowedNssais\": null, \"priority\": 4, \"capacity\": 50, \"load\": 10,
\"recoveryTime\": 1542876663222, \"supportedFeatures\": null}]},
{\"nfInstanceId\": \"fe7d992b-0541-4c7d-ab84-555554444444\", \"nfSetIdList\":
[\"set002.udrset.5gc.mnc012.mcc345\"], \"nfType\": \"UDR\", \"nfStatus\":
\"REGISTERED\", \"plmnList\": null, \"nsiList\": null, \"fqdn\":
\"nf12stub.gi-dina.svc\", \"interPlmnFqdn\": null, \"ipv4Addresses\": null,
\"ipv6Addresses\": null, \"priority\": 5, \"capacity\": 50, \"load\": 40,
\"locality\": \"2\", \"pcfInfo\": null, \"udmInfo\": null, \"ausfInfo\":
null, \"amfInfo\": null, \"smfInfo\": null, \"upfInfo\": null, \"groupId\":
\"udr1\", \"udrInfo\": {\"supiRanges\": [{\"start\": \"450081000000000\",
\"end\": \"450081002000000\"}], \"gpsiRanges\": [{\"start\": \"13100000000\",
\"end\": \"13102000000\"}], \"supportedDataSets\": [\"POLICY\"]},
\"bsfInfo\": null, \"customInfo\": null, \"recoveryTime\": null,
\"nfServices\": [{\"serviceInstanceId\": \"94d8d19a-
b0d9-4d6d-994f-39a49ed5c111\", \"serviceName\": \"nudr-dr\", \"versions\":
[{\"apiVersionInUri\": \"v1\", \"apiFullVersion\": \"1.15.1.0\", \"expiry\":
\"2019-08-03T18:55:08.871+0000\"}], \"scheme\": \"http\",
\"nfServiceStatus\": \"REGISTERED\", \"fqdn\": null, \"interPlmnFqdn\": null,
\"ipEndPoints\": [{\"ipv4Address\": null, \"ipv6Address\": null,
\"transport\": \"TCP\", \"port\": 8080}], \"apiPrefix\": null,
\"defaultNotificationSubscriptions\": null, \"allowedPlmns\": null,
\"allowedNfTypes\": [\"CHF\", \"PCF\"], \"allowedNfDomains\": null,
\"allowedNssais\": null, \"priority\": 5, \"capacity\": 50, \"load\": 40,
\"recoveryTime\": 1542876663222, \"supportedFeatures\": null}]},
{\"nfInstanceId\": \"fe7d992b-0541-4c7d-ab84-555552222222\", \"nfSetIdList\":
[\"set002.udrset.5gc.mnc012.mcc345\"], \"nfType\": \"UDR\", \"nfStatus\":
\"REGISTERED\", \"plmnList\": null, \"nsiList\": null, \"fqdn\":
\"nf22stub.gi-dina.svc\", \"interPlmnFqdn\": null, \"ipv4Addresses\": null,
\"ipv6Addresses\": null, \"priority\": 3, \"capacity\": 50, \"load\": 30,
\"locality\": \"1\", \"pcfInfo\": null, \"udmInfo\": null, \"ausfInfo\":
null, \"amfInfo\": null, \"smfInfo\": null, \"upfInfo\": null, \"groupId\":
\"udr1\", \"udrInfo\": {\"supiRanges\": [{\"start\": \"450081000000000\",
\"end\": \"450081002000000\"}], \"gpsiRanges\": [{\"start\": \"13100000000\",
\"end\": \"13102000000\"}], \"supportedDataSets\": [\"POLICY\"]},
\"bsfInfo\": null, \"customInfo\": null, \"recoveryTime\": null,
\"nfServices\": [{\"serviceInstanceId\": \"94d8d19a-
b0d9-4d6d-994f-39a49ed5c111\", \"serviceName\": \"nudr-dr\", \"versions\":
[{\"apiVersionInUri\": \"v1\", \"apiFullVersion\": \"1.15.1.0\", \"expiry\":
\"2019-08-03T18:55:08.871+0000\"}], \"scheme\": \"http\",
\"nfServiceStatus\": \"REGISTERED\", \"fqdn\": null, \"interPlmnFqdn\": null,

\"ipEndPoints\": [{\"ipv4Address\": null, \"ipv6Address\": null,
\"transport\": \"TCP\", \"port\": 8080}], \"apiPrefix\": null,
\"defaultNotificationSubscriptions\": null, \"allowedPlmns\": null,
\"allowedNfTypes\": [\"CHF\", \"PCF\"], \"allowedNfDomains\": null,
\"allowedNssais\": null, \"priority\": 3, \"capacity\": 50, \"load\": 30,
\"recoveryTime\": 1542876663222, \"supportedFeatures\":
null}]}]}","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogg
er","threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.2
03+0000"}
{"instant":
{"epochSecond":1702448069,"nanoOfSecond":203984193},"thread":"XNIO-1
task-2","level":"INFO","loggerName":"com.oracle.cgbu.cnc.nrf.core.usecases.nrf
.GetNrfDiscoveryResponseUseCase","message":"Dynamic discovery cache enabled :
false","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger",
"threadId":90,"threadPriority":5,"messageTimestamp":"2023-12-13T06:14:29.203+0
000"}

4.23 Handling Race Condition Between Gx and Sy Sessions in two sites

Policy supports handling race condition between Gx and Sy sessions in two sites that result in Sy stale sessionsin PDS.

Race condition between Gx and Sy sessions

A sample race condition is when the PGW opens two different sessions (such as data and voice or data and data sessions) with two different PCRF Core instances on site1 and site2, termination request for both the sessions are sent to different sites.

While the session create Diameter Credit-Control-Request (CCR)-I appears to come in a sequence succession, first data and then voice, the terminate request for both voice and data are received at the same time in two different sites, while the device disconnects from the network.

As this is happening at the same time, both sites Site 1 and Site 2 in their local database find that there is another Gx session open. Both sites close their own Gx sessions, but as they both (at this time) see a remaining Gx session at the other site, they keep the Sy session as open.

As the Session Termination Request (Sy-STR) is missing, the session remains active in OCS resulting in stale Sy session.

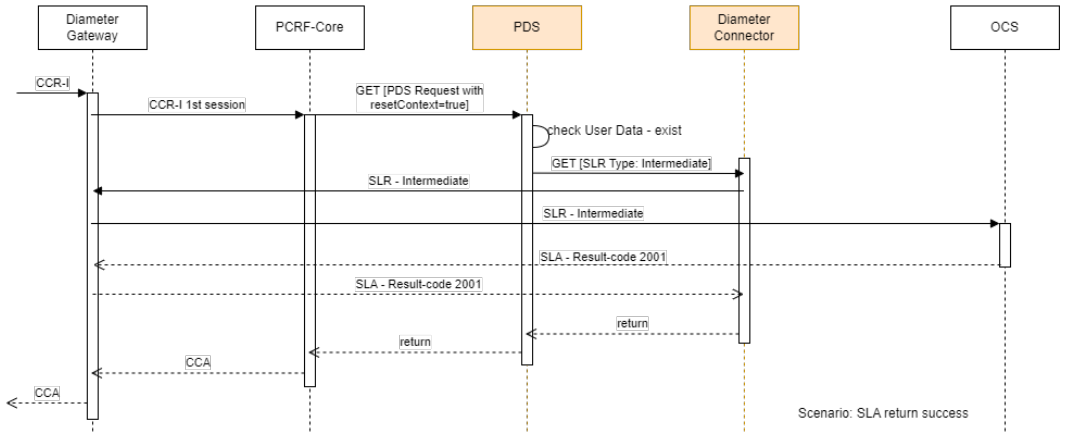
> **Note**
>
> Currently, Policy only supports minimizing the number of stale sessions, but cannot completely avoid the race condition.
>
> Also, Policy does not support error handling for this feature.

Handling Stale Sy sessions

Revalidating the Sy session to ensure that the Sy session is still valid by polling OCS (ResetContext)

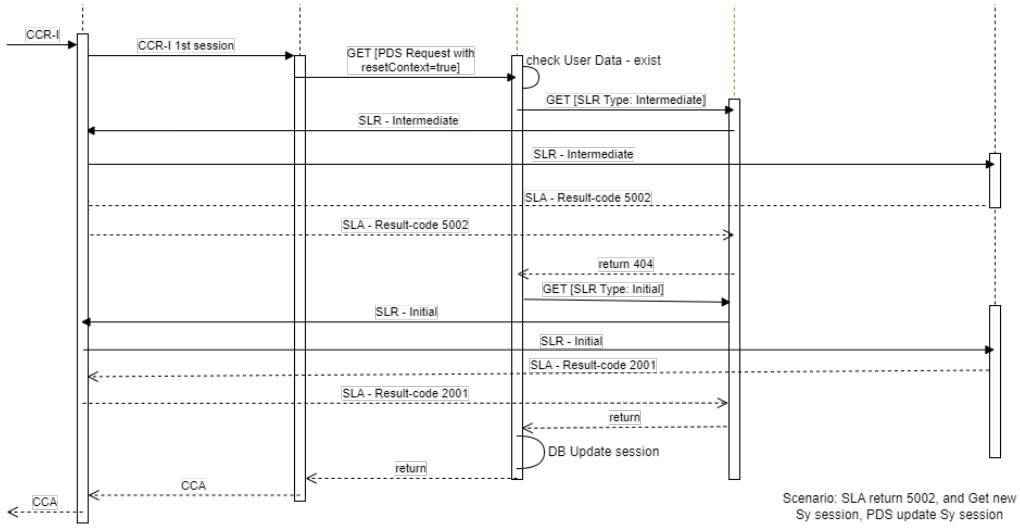
1. Upon establishment of the first Gx session for the subscriber, if the Sy data is present on PDS, PDS sends an Sy Spending-Limit-Request (SLR)-Intermediate to OCS requesting all the policy counters.
2. If PDS receives a success response from OCS, it indicates that PDS is in sync with OCS.
3. If PDS receives UNKNOWN_SESSION_ID error, PDS deletes the old Sy session and creates a new one.

Figure 4-24 SLA Returns Success



1. PCRF Core receives a CCR-I for first session through Diameter Gateway.
2. PCRF Core sends a Get request to PDS with `resetContext` flag set to `true`.
3. PDS searches for the corresponding user data in its database.
4. If the data exists, PDS sends an Sy SLR-Intermediate request to OCS to fetch all the policy counters. PDS sends this request through Diameter Connector routed through Diameter Gateway.
5. OCS reponds with Spending Limit Answer (SLA) with success result code 2001. The response is sent from OCS to Diameter Connector routed through Diameter Gateway.
6. Diameter Connectors returns the success response to PDS.
7. PDS updates the details in its database and returns the response to PCRF Core.
8. PCRF Core responds to the CCR-I request for session 1 with Diameter Credit-Control-Request (CCA).

Figure 4-25 SLA Returns 5002, Get New Sy Session, PDS Update Sy Session



1. PCRF Core receives a CCR-I for first session through Diameter Gateway.
2. PCRF Core sends a Get request to PDS with `resetContext` flag set to `true`.
3. PDS searches for the corresponding user data in its database.
4. If the data exists, PDS sends an Sy SLR-Intermediate request to OCS to fetch all the policy counters. PDS sends this request through Diameter Connector routed through Diameter Gateway.
5. OCS reponds with Spending Limit Answer (SLA) with success result code 5002 indicating Diameter Unknown Session Id. The response is sent from OCS to Diameter Connector routed through Diameter Gateway.
6. Diameter Connector sends a 404 error to PDS.
7. PDS sends a GET request with SLR Type: Initial to Diameter Connector to get the new Sy session.
8. Diameter Connector forwards the SLR - Initial request to OCS through Diameter Gateway.
9. OCS sends SLA with Result-Code 2001 to Diameter Connetor through Diameter Gateway indicating Diameter Success.
10. Diameter Connector returns the success response to PDS.
11. PDS updates its database and returns the response to PCRF Core.
12. PCRF Core responds to the CCR-I request for session 1 with Diameter Credit-Control-Request (CCA).

> **Note**
>
> For any unsuccessful response other than 5002 (404) the existing record will be deleted.

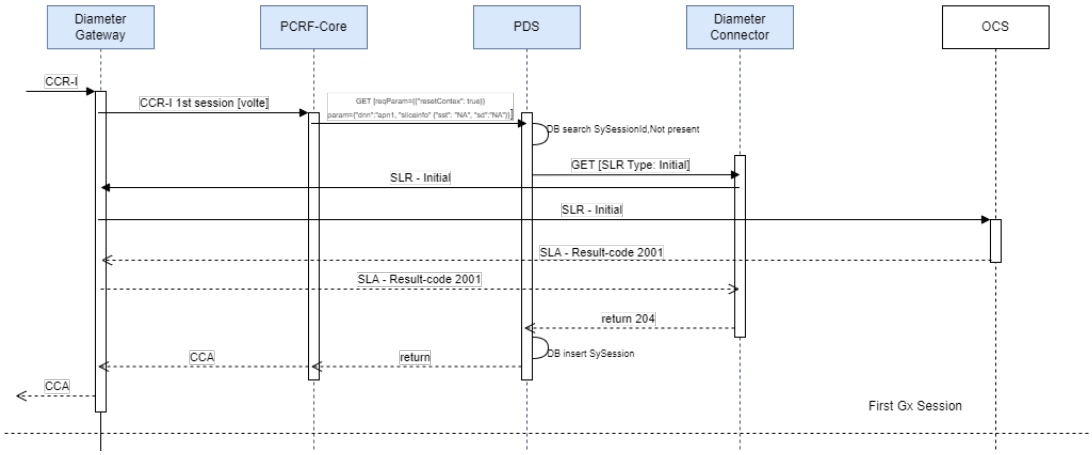
Handling stale Sy sessions by limiting the number of Gx sessions that can be associated with an Sy session for a specific APN

1. PCRF provides **Max Sessions Count** to specify how many Gx sessions can be associated with an Sy session for a specific APN.
2. If **Max Sessions Count** exceeds the limit, PCRF revalidates the Sy session with OCS.

This method protects against cases where Gx CCR-T are not sent by the PGW/received by the PCRF for a specific APN.

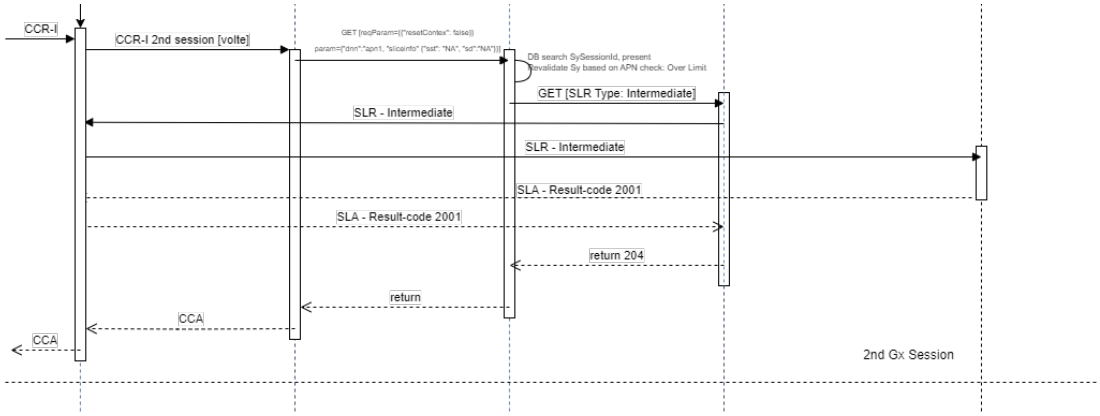
Here is an example call flow with **Max Sessions Count**=1. The request for the first Gx session results with a success response, while the second request is revalidated.

Figure 4-26 First Gx Session



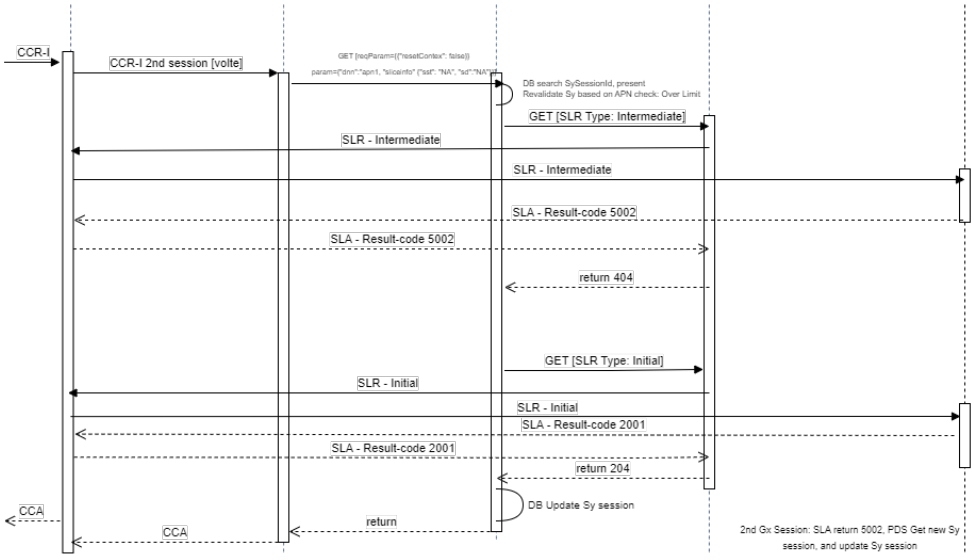
1. PCRF Core receives a CCR-I for first session through Diameter Gateway.
2. PCRF Core sends a Get request to PDS with `resetContext` flag set to `true` along with other parameters such as DNN (with Data Network Name), SST (with Slice or Service Type) and SD (with Slice Differentiator Name).
3. PDS searches for the data in its database.
4. If the data does not exist for the given Sy SessionId, PDS sends an Sy SLR-Intermediate request to OCS to fetch all the policy counters. PDS sends this request through Diameter Connector routed through Diameter Gateway.
5. OCS reponds with Spending Limit Answer (SLA) with success result code 2001. The response is sent from OCS to Diameter Connector routed through Diameter Gateway.
6. Diameter Connectors returns the success response to PDS.
7. PDS updates the details in its database and returns the response to PCRF Core.
8. PCRF Core responds to the CCR-I request for session 1 with Diameter Credit-Control-Request (CCA).

Figure 4-27 Second Gx Session



1. PCRF Core receives a CCR-I for second session through Diameter Gateway.
2. PCRF Core sends a Get request to PDS with `resetContext` flag set to `false` along with other parameters such as DNN (with Data Network Name), SST (with Slice or Service Type) and SD (with Slice Differentiator Name).
3. PDS searches for the data in its database.
4. If the data exists for the given Sy SessionId, PDS sends an Sy SLR-Intermediate request to OCS to fetch all the policy counters. PDS sends this request through Diameter Connector routed through Diameter Gateway.
5. OCS reponds with Spending Limit Answer (SLA) with success result code 2001. The response is sent from OCS to Diameter Connector routed through Diameter Gateway.
6. Diameter Connectors returns 204 no data response to PDS.
7. PDS updates the details in its database and returns the response to PCRF Core.
8. PCRF Core responds to the CCR-I request for session 2 with Diameter Credit-Control-Request (CCA).

Figure 4-28 Second Gx Session: SLA Returns 5002, Get New Sy Session, PDS Update Sy Session



1. PCRF Core receives a CCR-I for second session through Diameter Gateway.
2. PCRF Core sends a Get request to PDS with `resetContext` flag set to `false` along with other parameters such as DNN (with Data Network Name), SST (with Slice or Service Type) and SD (with Slice Differentiator Name).
3. PDS searches for the data in its database.
4. If the data exists for the given Sy SessionId, PDS sends an Sy SLR-Intermediate request to OCS to fetch all the policy counters. PDS sends this request through Diameter Connector routed through Diameter Gateway.
5. OCS reponds with Spending Limit Answer (SLA) with success result code 5002. The response is sent from OCS to Diameter Connector routed through Diameter Gateway.
6. Diameter Connectors returns 404 error to PDS.
7. PDS sends a GET request with SLR Type: Initial to Diameter Connector to get the new Sy session.
8. Diameter Connector forwards the SLR - Initial request to OCS through Diameter Gateway.
9. OCS sends SLA with Result-Code 2001 to Diameter Connetor through Diameter Gateway indicating Diameter Success.
10. Diameter Connector returns the 204 response to PDS.
11. PDS updates its database and returns the response to PCRF Core.
12. PCRF Core responds to the CCR-I request for session 1 with Diameter Credit-Control-Request (CCA).

Managing Handling Race Condition Between Gx and Sy Sessions over two sites

Enable

To enable the PDS revalidation functionality:

You can enable the PDS revalidation functionality using the CNC Console or REST API for Policy.

- **Enable using CNC Console**:
  To enable revalidation of the Sy stale sessions, set the value of `USER.ocsSpendingLimit.resetContextOnGxCreate` key under **Advanced Settings** in **Settings** page for **PCRF Core** to `true`.

  For more details, see PDS Settings.
- **Enable using REST API**:
  To enable the revalidation functionality, configure the advanced settings keys in the `{apiRoot}/oc-cnpolicy-configuration/v1/services/pcrfcore/settings` REST API.

  For more details, see *Policy REST Specifications* section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

You can configure the revalidation functionality to handle the stale Sy sessions using CNC Console or REST API for Policy.

- **Configure using CNC Console**:
  For revalidating Sy sessions, configure **Enable Fetch and Resubscribe** and **Session Count per DNN/APN List Settings** under **Settings** page for **PDS**.

  For more information, see PDS Settings.

- **Configure using REST API**: Policy provides {apiRoot}/oc-cnpolicy-configuration/v1/services/pds/pdsSettings REST API to configure the PDS revalidation functionality.

Observability

Metrics

The following PDS metrics are used to revalidate the Sy sessions:

- revalidation_request
- revalidation_response

For more details on these metrics, see Policy DS Metrics.

4.24 Support of Policy Action to Send the Notify Terminate

PCF provide policy rules to network functions SMF/AMF. It integrates with AMF for Access and Mobility Policy Control and User Equipment (UE) related policies and with SMF for session management policies. To provide updated Policies, PCF initiates an `update` notification request toward SMF/AMF. On receiving SMF/AMF response, the PCF either decides to update the Policies or sends request to terminate the policy association.

The PCF policy engine evaluates configured Policies that are triggered by events received from SMF, AMF, CHF, and UDR. The policy triggers are sent to PCF during the following cases:

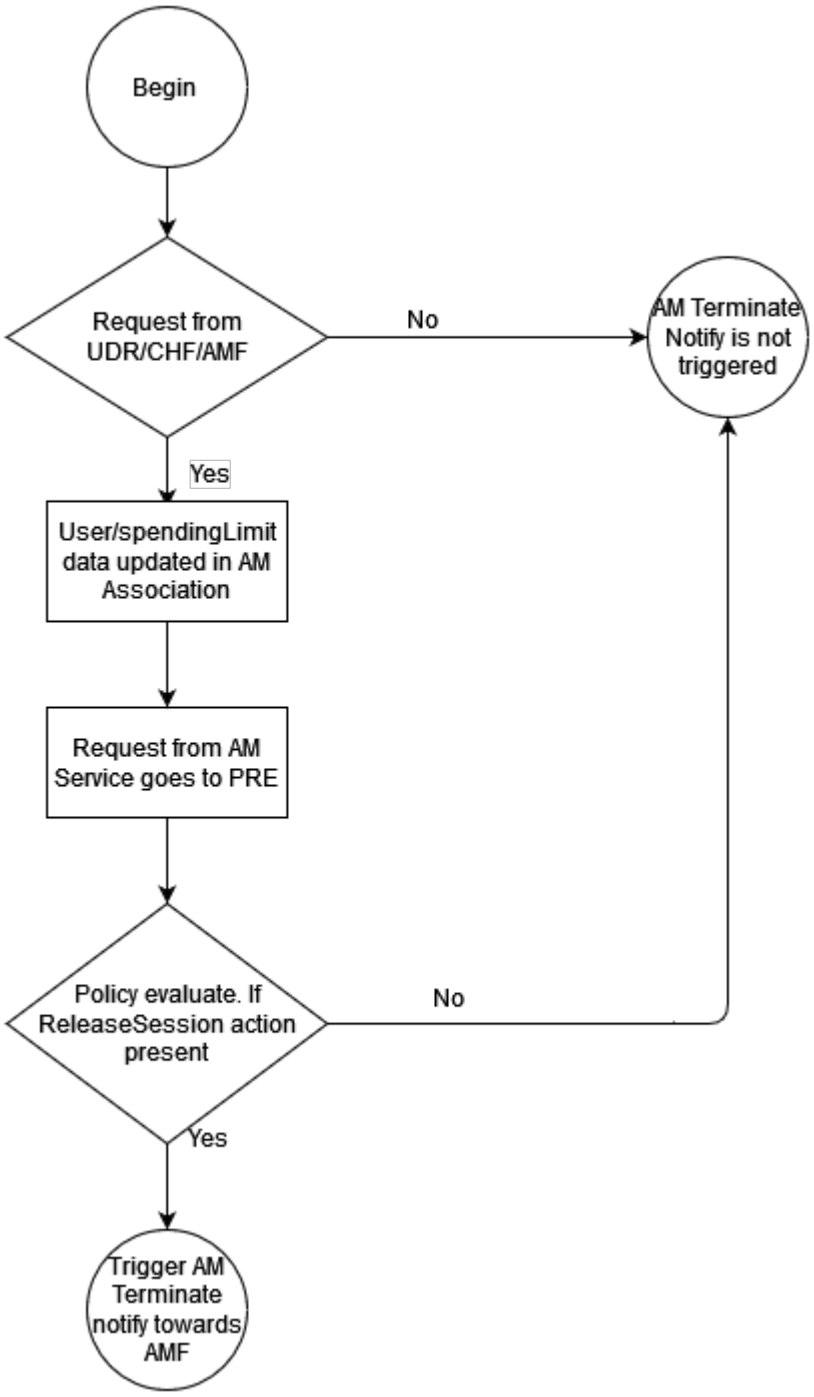
- UDR notifies the PCF about a Policy subscription changes
- CHF notifies PCF of Policy counter status changes
- AMF notifies PCF for AM and UE Policy updates
- SMF notifies PCF for session and PCC updates

On receiving policy trigger information, the Policy Runtime Engine (PRE) evaluates the Policies and makes the requested policy decision. The PCF may decide to send an update or terminate request toward AMF/SMF. If PCF decides to terminate, it invokes `Npcf_AMPolicyControl_UpdateNotify` service operation toward AMF or `Npcf_SMPolicyControl_UpdateNotify` toward SMF requesting for termination of the policy association.

PCF Triggering AM UpdateNotify Terminate

The following flowchart illustrates the flow of PCF triggering AM terminateNotify request toward AMF for termination of the Policy association.

Figure 4-29 PCF Triggering AM UpdateNotify Terminate



1: Start

2: If notifications requests are coming from either UDR or CHF or AMF, go to step 3, if not, go to step 7.

3: User/Spending limit data updated in AM association.

4: Request from AM service goes to PRE.

5: PRE evaluates the policies. If ReleaseSession action (with or without cause) is present go to step 6, if not, go to step 7.

6: Trigger AM terminate notify toward AMF.

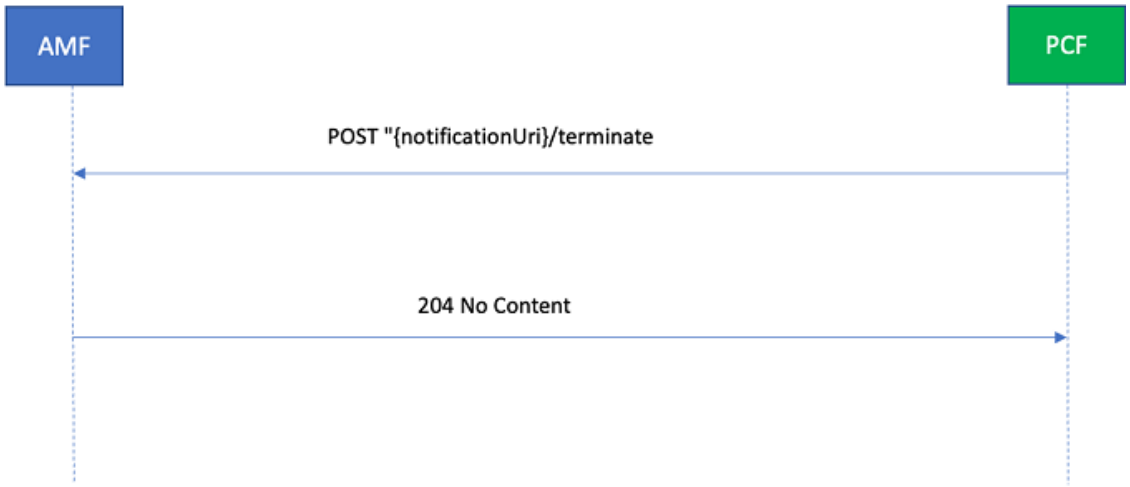
7: AM Terminate notify is not triggered.

If PCF requests the termination of AM Policy association, it sends an HTTP Post request with "{notificationUri}/terminate" as URI (where the notification URI was previously supplied by the AMF). The request body of termination notification includes:

- the policy association Id encoded as "polAssoId" attribute; and
- the cause as to why PCF requests the termination of the policy association encoded as "cause" attribute

The following figure, illustrates the request for termination of the policy association from PCF.

Figure 4-30 Deletion of Policy Association



PCF termination request is made with `TerminationNotification` data type.

Table 4-13 Definition of Type TerminationNotification

Attribute Name	Data Type	Description
resourceUri	Uri	The resource URI of each AM/UE/SM policy association related to the notification.
cause	PolicyAssociationReleaseCause	The cause for PCF requesting for Policy association termination.

Table 4-14 Enumeration PolicyAssociationReleaseCause

Enumeration value	Description
UNSPECIFIED	This value is used for unspecified reasons.

Table 4-14 (Cont.) Enumeration PolicyAssociationReleaseCause

Enumeration value	Description
UE_SUBSCRIPTION	This value indicates that the policy association needs to be terminated because the subscription of UE has changed such as removal of a subscription.
INSUFFICIENT_RES	This value indicates that the server is overloaded and needs to terminate the policy association.
REACTIVATION_REQUESTED This enumeration value is exclusive to **SmPolicyAssociationReleaseCause** only.	This value indicates that policy association needs to be terminated, since PCF is not able to maintain the existing PDU session. PCF requests for PDU session reactivation.

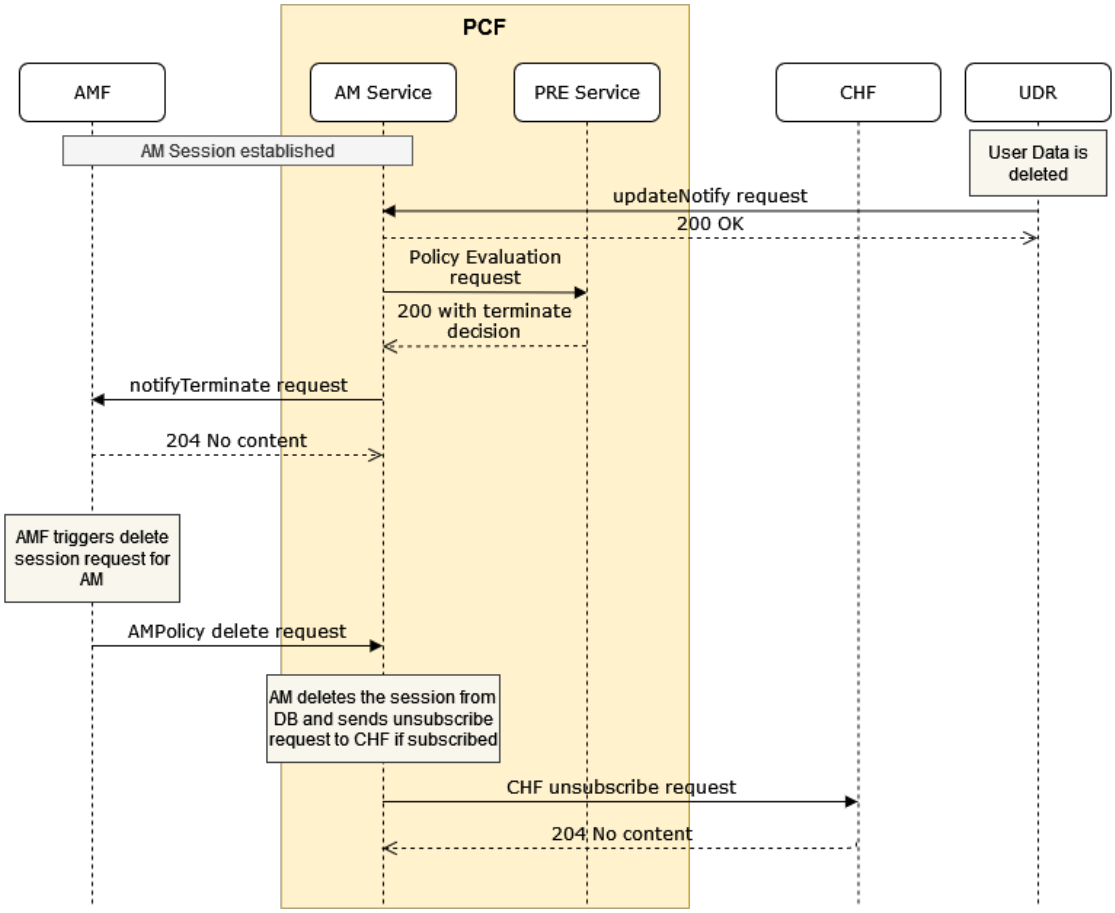
Policy blockly `Release Session with cause` holds the PCF cause for termination of AM policy association. The cause for termination is configurable by the user and the default cause value is `UE_SUBSCRIPTION`.

AMF on receiving post request for termination from PCF, either sends a *204 No Content* response for the successful processing or an appropriate failure response. After successful processing of PCF request, AMF invokes `Npcf_AMPolicyControl_Delete` service operation to terminate the policy association. Upon receiving the delete request, PCF deletes the Policy association and its associated sessions in UDR. It sends either a `204 No Content` response indicating the success of the deletion or an appropriate failure response.

When PCF does not receive the delete request from AMF, the Policy association is not deleted in Policy and thus its associated resources with the external NFs are not deleted.

AM Service Call Flow

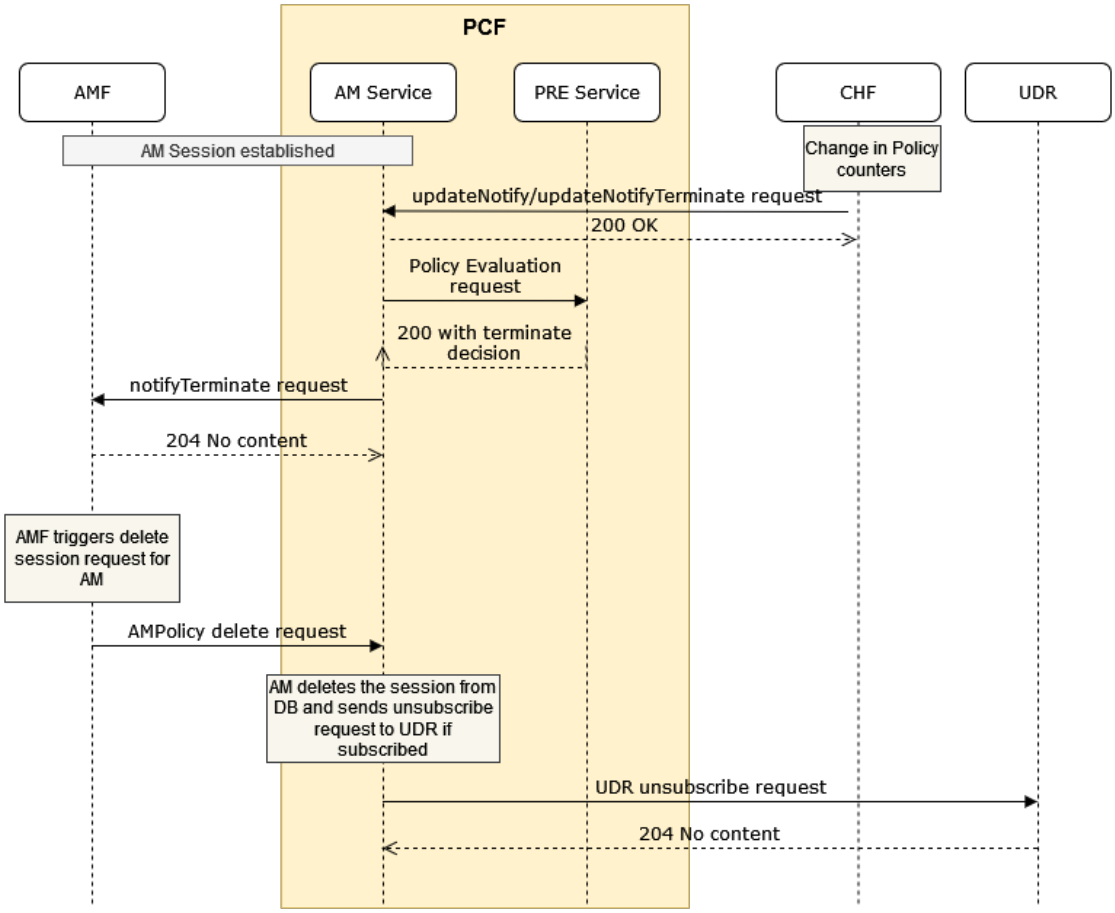
Figure 4-31 Call Flow for AM Terminate for UDR updateNotify with delResources



1. AMF sends a UE session established message to AM Policy service.
2. At UDR when a user data is deleted then, UDR sends an *updateNotify* request to AM service. The updateNotify request includes the `delResources` property containing the path of the resource that was deleted.
3. AM Policy service forwards the `200 OK` message to UDR.
4. AM Policy service sends a *Policy evaluation* request to PRE to evaluate the details.
5. Depending on the evaluation, PRE includes `Release Session` action in its response to AM Policy service.
6. In case of termination, AM Policy service sends a *terminateNotify* request to AMF.
7. AMF sends a `204 No Content` response to AM Policy service indicating that the request was successful.
8. AMF triggers a delete session requests and sends *AM Policy delete* request to AM Policy service.
9. AM Policy service deletes the AMPolicyAssociation from the database and sends an *unsubscribe* request to CHF.

10. CHF removes the subscription and responds to AM Policy service with `204 No Content` message.

Figure 4-32 Call Flow for AM Terminate for CHF updateNotify or TerminateNotify

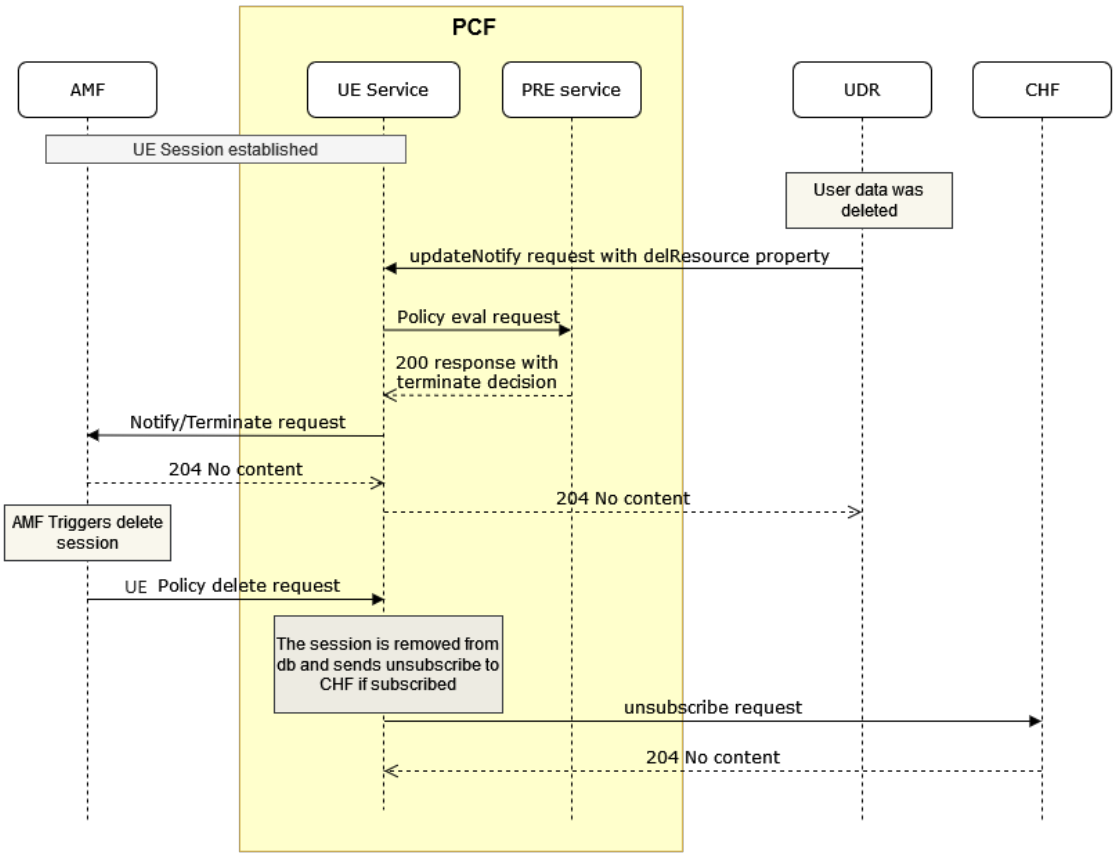


1. AMF sends an AM session established message to AM Policy service.
2. When the Policy Counters for a subscriber is deleted in CHF, it sends an *updateNotify* or*terminateNotify* request to AM Policy service.
3. AM Policy service forwards the `200 OK` message to CHF.
4. AM Policy service sends a *Policy evaluation* request to PRE to evaluate the details.
5. Depending on the evaluation, PRE includes *Release Session* action in its response to AM Policy service.
6. AM Policy service sends a *notifyTerminate* request to AMF.
7. AMF sends a `204 No Content` response to AM Policy service indicating that the request was successful.
8. AMF triggers a delete session request and sends a AMPolicy delete request to AM Policy service.
9. AM Policy service deletes the AMPolicyAssociation from the database and sends an *unsubscribe* request to UDR, if subscribed.

10. UDR removes the subscription and responds to AM Policy service with `204 No Content` message.

UE Service Call Flow

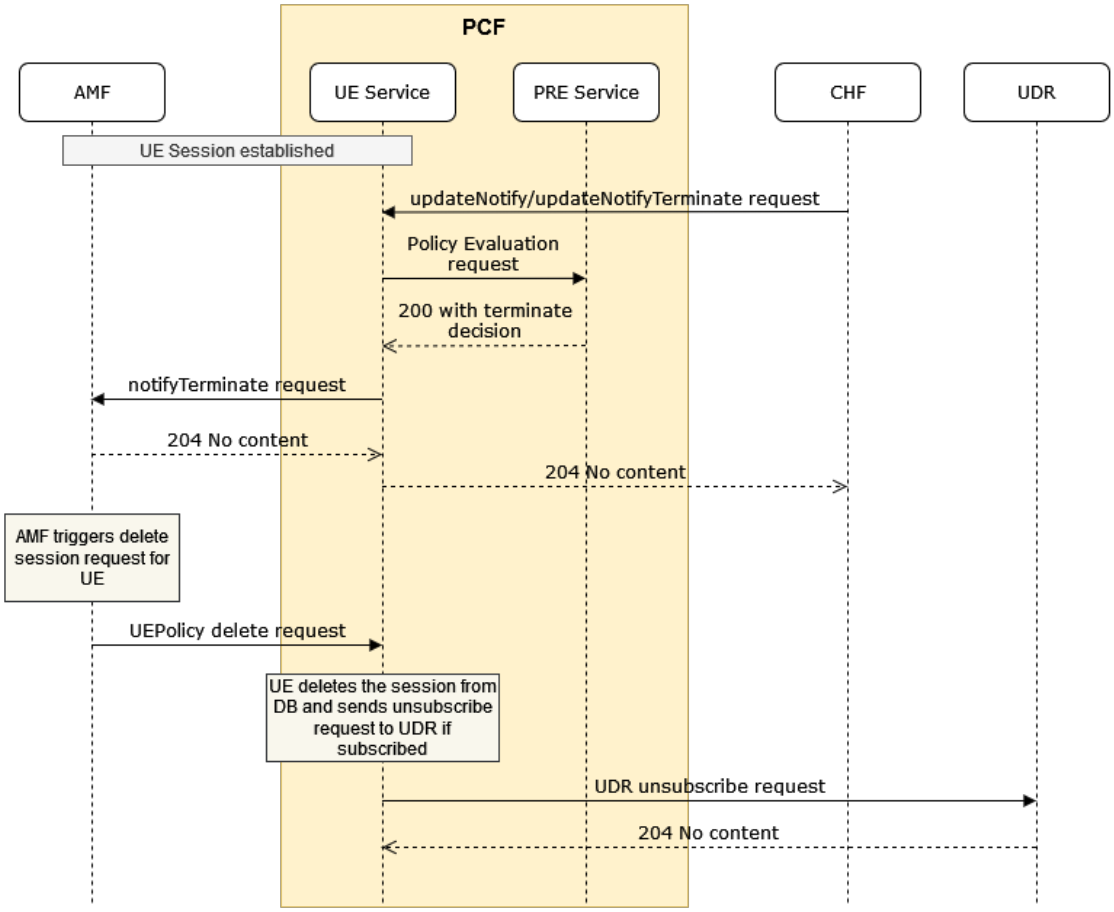
Figure 4-33 Call Flow for UE Terminate for UDR updateNotify with delResources



1. AMF sends a UE Session established message to UE Policy service.
2. When a user data is deleted in UDR, UDR sends an *updateNotify* request to UE Policy service. The updateNotify request includes the *delResources* property, containing the path of the resource that was deleted.
3. UE Policy service sends a *Policy evaluation* request to PRE to evaluate the details.
4. Depending on the evaluation, PRE includes *Release Session* action in its response to UE Policy service.
5. UE Policy service sends a *notifyTerminate* request to AMF.
6. AMF sends a `204 No Content` response to UE Policy service indicating that the request was successful.
7. UE Policy service forwards the `204 No Content` message to UDR.
8. AMF triggers a delete session request and sends a UEPolicy delete request to UE Policy service.
9. UE Policy service deletes the UEPolicyAssociation from the database and sends an `unsubscribe` request to CHF, if subscribed.

10. CHF removes the subscription and responds to UE Policy service with `204 No Content` message.

Figure 4-34 Call Flow for UE Terminate for CHF updateNotify or updateNotifyTerminate

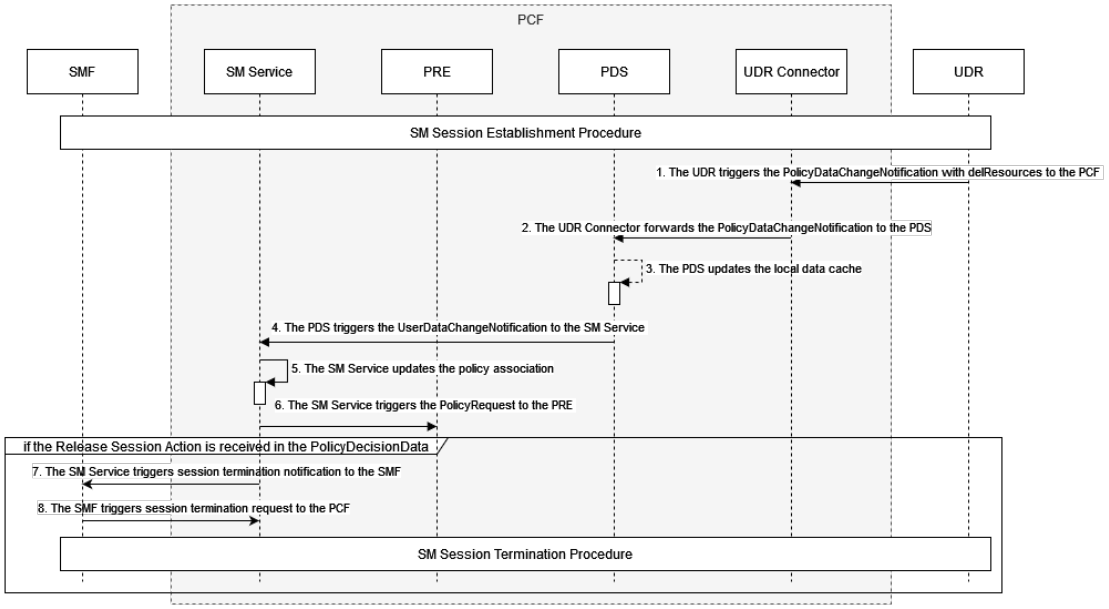


1. AMF sends a UE Session established message to UE Policy service.
2. When the Policy Counters for a subscriber is deleted in CHF, CHF sends an *updateNotify* or *updateNotify Terminate* request to UE Policy service.
3. UE Policy service sends a *Policy evaluation* request to PRE to evaluate the details.
4. Depending on the evaluation, PRE includes *Release Session* action in its response to UE Policy service.
5. UE Policy service sends a *notifyTerminate* request to AMF.
6. AMF sends a `204 No Content` response to UE Policy service indicating that the request was successful.
7. UE Policy service forwards the `204 No Content` message to CHF.
8. AMF triggers a delete session request and sends a UEPolicy delete request to UE Policy service.
9. UE Policy service deletes the UEPolicyAssociation from the database and sends an *unsubscribe* request to UDR, if subscribed.

10. UDR removes the subscription and responds to UE Policy service with `204 No Content` message.

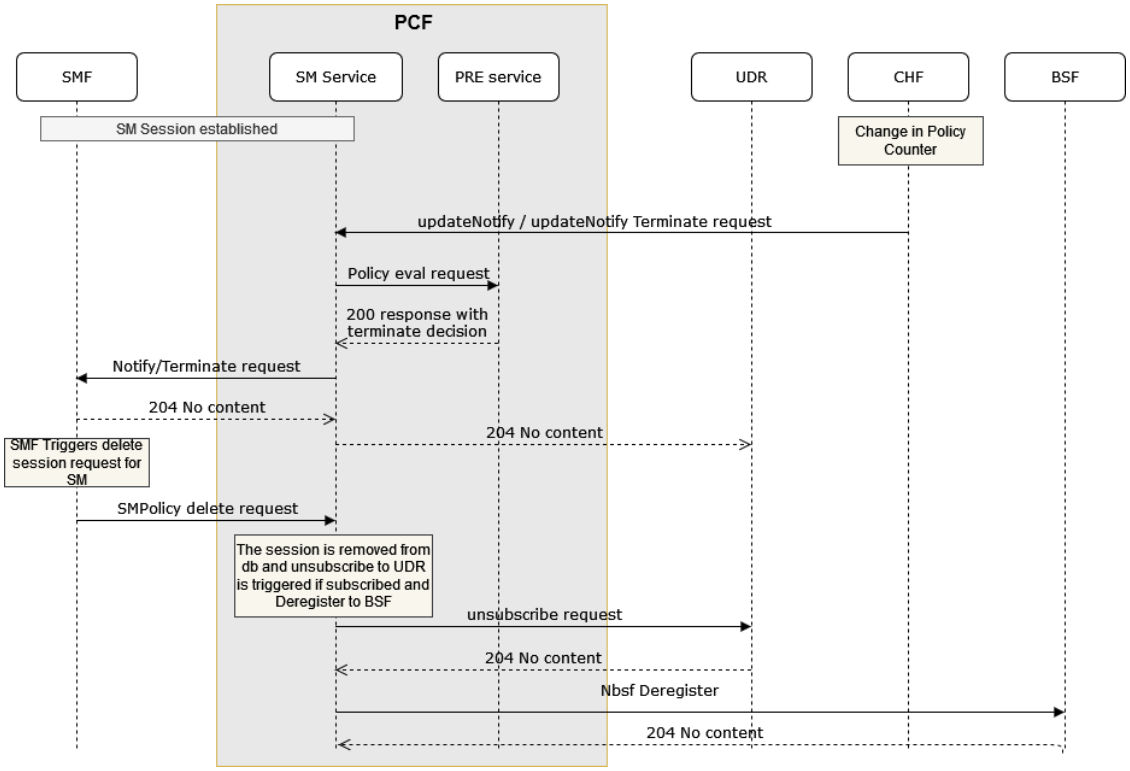
SM Service Call Flow

Figure 4-35 Call Flow for SM Terminate for UDR updateNotify with delResources



1. The UDR triggers the PolicyDataChangeNotification with the attribute "delResources" to the PCF.
2. The UDR Connector forwards the received PolicyDataChangeNotification to the PDS.
3. After receiving the UDR notification PDS updates the database
4. PDS triggers the user data change notification request to SM service.
5. The SM service updates the Policy association.
6. The SM service sends policy request to PRE.
7. If PRE decides to terminate SM policy association, SM service triggers session termination notification to the SMF.
8. The SMF triggers session termination request to the PCF

Figure 4-36 Call Flow for AM Terminate for CHF updateNotify or TerminateNotify



1. SMF sends a SM session established message to SM Policy service.
2. When the Policy Counters for a subscriber is deleted in CHF, CHF sends an *updateNotify* or *updateNotify Terminate* request to SM Policy service.
3. SM Policy service sends a *Policy evaluation* request to PRE to evaluate the details.
4. Depending on the evaluation, PRE includes *Release Session* action in its response to SM Policy service.
5. SM Policy service sends a *notifyTerminate* request to SMF.
6. SMF sends a `204 No Content` response to SM Policy service indicating that the request was successful.
7. SM Policy service forwards the `204 No Content` message to CHF.
8. SMF triggers a delete session request and sends a SMPolicy delete request to SM Policy service.
9. SM Policy service deletes the SMPolicyAssociation from the database, sends an *unsubscribe* request to UDR, if subscribed and *deregisteration* request to BSF.
10. UDR removes the subscription and responds to SM Policy service with `204 No Content` message.
11. BSF deregisters the subscriber and responds to SM Policy service with `204 No Content` message.

PCF Decision Not to Terminate AM/SM Policy Association

In case of the PCF decision, to keep the Policy associations and not to terminate it, the default policy is used and data is not fetched from the UDR.

If PCF decides to keep the Policy associations, the default policy is used and not the data from UDR. PCF invokes UpdateNotify service operation toward AMF/SMF to update the AM/SM Policy control information. PCF updates the local database as ampolicydata/smpolicydata. Since the data from UDR is not present for the subscriber, UDR should not have any subscription for this subscriber.

When PCF receives updates of an Policy association from the AMF/SMF for this session, it performs UDR query request based on configurations from `am-data/sm-data`. If UDR sends an error response due to USER_NOT_FOUND, then PCF takes actions according to "Error handling feature for AM/SM Policy" feature.

Managing Support of Policy Action to Send the Notify Terminate

This section explains the procedure to enable and configure the feature.

Enable

This forms Policy applications core feature functionality. You do not need to enable or disable this feature.

Configure Using Blockly

New blocks `UDR delResources contains`, `Release Session with cause` and `Release session` without cause are introduced in AM/UE/SM services. For more information, see *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.

Observability

Metrics:

The following AM service metrics are used for this feature:

- `ocpm_egress_request_total`
- `ocpm_egress_response_total`

The following UE service metrics are used for this feature:

- `http_in_conn_request`
- `http_out_conn_response`

UDR service uses `ocpm_userservice_inbound_count_total` metric for this feature.

Maintain
If you encounter alerts at system or application levels, see Alerts section for resolution steps.

In case the alerts still persist, perform the following:

- Collect the logs: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

4.25 Bulwark Pod Congestion Control

Bulwark, a Policy microservice provides distributed lock mechanism using distributed coherence. It handles concurrent transactions across consumer services. Since the lock is distributed, the deployed pods request a lock for a resource based on a unique identifier in the global system rather than acquiring a lock locally in a single pod.

The traffic at bulwark service is high as all the consumer services use its distributed lock mechanism to handle concurrent transactions. In the production environment, the bulwark pods must be protected from traffic congestion. Bulwark pod congestion control functionality helps in regulating the traffic and improves its service availability.

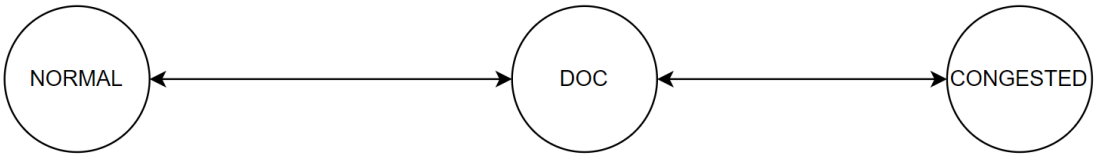
Determining Pod Congestion State

Bulwark pod exists in any of the following three states at any given time:

1. Normal
2. DOC (Danger of Congestion)
3. Congested

Periodically, the state of the pod's congestion gets determined. This interval is configurable, and the default setting is 5000 milliseconds.

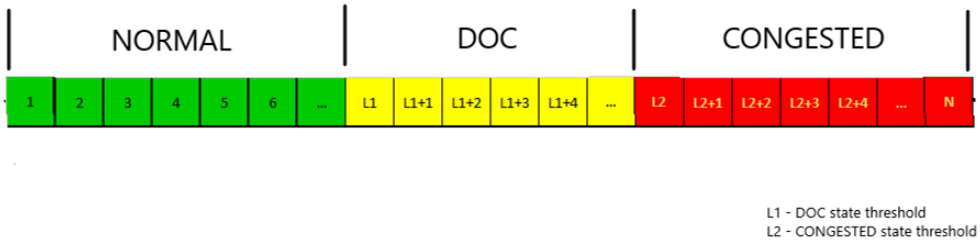
Figure 4-37 Different Pod Congestion States



The pod's state gets determined by considering the following points.

1. Calculate the congestion state for the following resources:
   a. **Queue**: For the DOC and CONGESTED pod states, compare the number of pending messages in the queue with the configured pending messages threshold.

Figure 4-38 Congestion States



   b. **CPU** - The CPU usage for congestion state is calculated by comparing the CPU usage of the Container (monitored using cgroup parameter - `cpuacct.usage` that provides current cpu usage in nanoseconds) with the configured threshold. The following formula calculates the CPU usage:

$$\frac{\frac{CurrentCpuUsage - LastCpuUsage}{CurrentTime - LastSampleTime} * 100}{CPUs}$$

c. **Memory** - In order to phase out memory-based congestion control in the future, the threshold for memory consumption for the congestion state is set at 100%.

2. The congestion state for pod gets assigned a maximum congested state based on the congestion state of the resources. For Example, the calculated congestion state of the pod becomes CONGESTED when its CPU becomes NORMAL and Queue becomes CONGESTED.
The following table provides the evaluated pod states for different CPU and Queue states:

Table 4-15 Published Pod Congestion State

Pod	CPU	Queue
CONGESTED	NORMAL	CONGESTED
DOC	DOC	NORMAL
DOC	DOC	DOC

3. The current congestion state of the pod holds the published state of the pod congestion state. This changes to the calculated congestion state only when the calculated state remains same for all the configured number of continuous sample counts. By doing so, the pod avoids events like short bust of traffic triggering a change in the congestion state and load shedding. However, in the following scenario, the current congestion state changes to DOC whenever:
   - the current state is NORMAL and calculated state is CONGESTED, or
   - the current state is CONGESTED and calculated state is NORMAL

Triggering Congestion Control

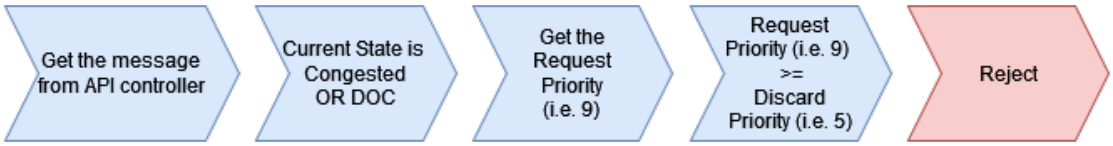
Every time Bulwark receives request to its lock and unlock service, the system checks for the current congestion state of the pod. The current congestion state on being Congested or DOC the congestion control mechanism gets triggered.

The lock and unlock requests have `oc-message-priority` attribute in the request header. The priority value ranges between 0 to 100 with 0 being the highest and 100 being the lowest priority.

The consumer services such as SM, AM, UE, and PDS services shall support priority header functionality to their lock and unlock requests based on message types and use cases.

Below figure illustrates this process.

Figure 4-39 Process Flow for Triggering Congestion Control



If the consumer services requests are without priority value set, then the bulwark service considers the default priority values. The Bulwark service default lock and unlock request priorities are:

- DEFAULT_LOCK_REQUEST_PRIORITY: 25
- DEFAULT_UNLOCK_REQUEST_PRIORITY: 15

Priority-Based Load Shedding

The congestion load rule configurations for current congestion state of the pod are applied to perform priority-based load shedding. It determines if the message with the assigned priority should be rejected or accepted.

These rules get configured per congestion state. If there are no rule configured for a congestion state, then bulwark accepts the request as a default behavior. The user can customize the result codes for the rejected requests when configuring the load rules. Customized the default result code is `503 Service Unavailable`.

The default load shedding rules for bulwark service:

```
- state: DANGER_OF_CONGESTION
      discardPriority: 20
- state: CONGESTED
      discardPriority: 10
```

When Bulwark pod is in congestion state, the response code for the rejected requests can be configured using the `responseCode` Helm parameter in the `values.yaml` file. By default, the `responseCode` parameter for Bulwark service is set to `500` response code. The user can configure this parameter with other supported 5xx response codes.

Sample Helm Configuration for Bulwark Service:

```
congestion:
     responseCode: 500
```

> **Note**
>
> In Policy **24.2.4**, the default response code is currently set to **500**.

SM Service Supporting the Priority Header for Lock and Unlock requests

SM service supports bulwark congestion control by adding the priority attribute `oc-message-priority` in the lock/unlock requests header. The 3gpp-Sbi-Message-Priority header is used to specify the message priority for 3GPP service based interfaces. This header is included in messages when a priority for the message must be conveyed. By default the oc-message-priority uses the 3gpp-Sbi-Message-Priority value to evaluate message priority. Requests without 3gpp-Sbi-Message-Priority header are assigned default values. The default priority values for the following requests are:

- CREATE - 24
- UPDATE - 18
- UPDATE_NOTIFY - 18
- DELETE - 16
- CLEANUP - 18

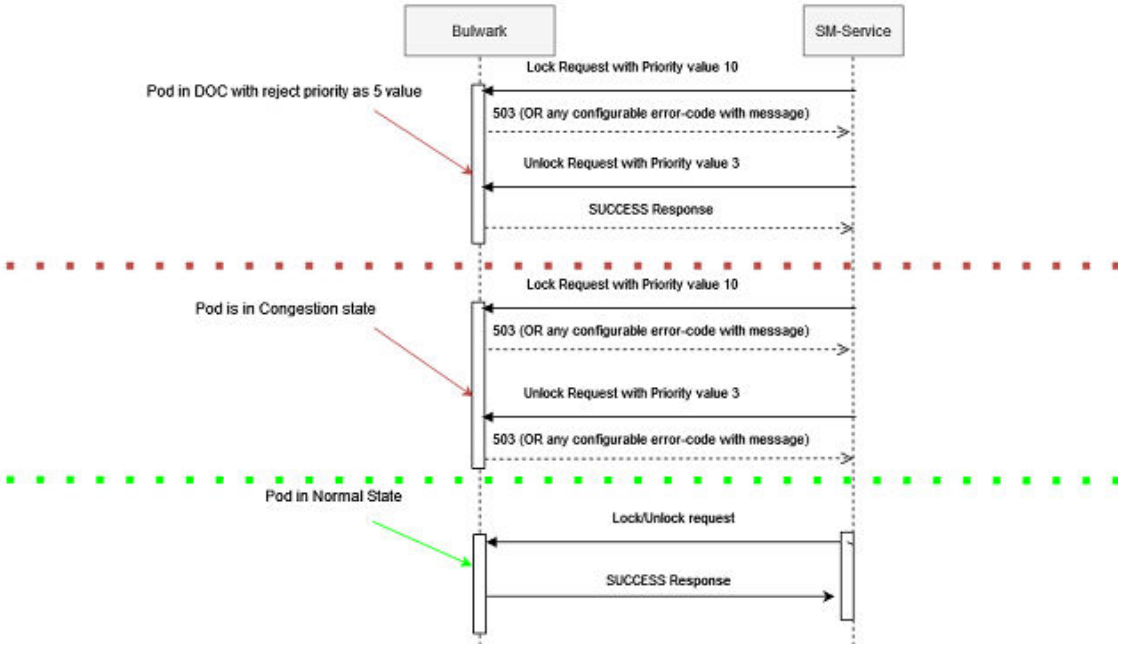
If the user wants to assign a different priority to the response message than the default ones, this can be configured using the Advanced settings of SM service in the CNC Console. Following are the advanced setting configurable parameters:

- `REQUEST_PRIORITY_FOR_SM_CREATE`
- `REQUEST_PRIORITY_FOR_SM_UPDATE`
- `REQUEST_PRIORITY_FOR_SM_UPDATE_NOTIFY`
- `REQUEST_PRIORITY_FOR_SM_DELETE`
- `REQUEST_PRIORITY_FOR_SM_CLEANUP`

Moreover, if the user does not want to consider the 3gpp-Sbi-Message-Priority value for oc-message-priority then the Advanced setting field `USE_TGPP_SBI_MSG_PRIORITY` should be set to false in the SM service configurations.

Below diagram shows the call flow for SM service supporting the priority header functionality.

Figure 4-40 Call Flow Diagram for Bulwark Congestion Control



- Bulwark receives an SM service lock request with priority set to 10 when it is in the DOC state with the reject priority set to 5. On comparing both priorities, bulwark rejects the message with result code since the message priority value is higher than the reject priority. The message returns with result code `503 Service unavailable` or any configurable error code with message.
- Bulwark receives an SM service unlock request with priority set to 3 when it is in the DOC state with the reject priority set to 5. On comparing both priorities, bulwark accepts the message with result code `Success` since the message priority value is lower than the reject priority.
- Bulwark receives an SM service lock request with priority set to 10 when it is in the CONGESTED state with reject priority set to 1. On comparing both priorities, bulwark rejects the message with result code since the message priority value is higher than the

reject priority. The message returns with result code `503 Service unavailable` or any configurable error code with message.

- Bulwark receives an SM service unlock request with priority set to 5 when it is in the CONGESTED state with reject priority set to 1. On comparing both priorities, bulwark rejects the message with result code since the message priority value is higher than the reject priority. The message returns with result code `503 Service unavailable` or any configurable error code with message.
- Bulwark receives an SM service lock or unlock request when it is in the NORMAL state. Bulwark accepts the message with result code `Success` since the requests do not have priority value set.

Managing Bulwark Pod Congestion Control

Enable

By default, the Pod Congestion control is disabled for Bulwark service. You can enable this feature using CNC Console or REST API for Policy.

> **Note**
>
> If user either adds or updates the Threshold configurations for Bulwark service, then Congestion Control feature gets disabled. The user will have to enable the congestion control feature again for Bulwark service using the **Settings** menu in CNC Console, and the new/updated Threshold configurations will be applied.

Configure Using CNC Console

Perform the Bulwark Congestion Control feature configurations on the Settings, Threshold and Load Shedding Rules in CNC Console as described in Congestion Control section.

Configure Using REST API

Perform the feature configurations as described in "Congestion Control" section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Observability

Metrics:

Following metrics were updated in the Pod Congestion Metrics section.

- `occnp_pod_congestion_state`
- `occnp_pod_resource_state`
- `occnp_pod_resource_congestion_state`

> **Note**
>
> Prometheus automatically injects name of the pod with label name "kubernetes_pod_name" to the metric. This information is further used for alerting purposes.

Alerts:

Following alerts are used by this feature:

- PodDoc
- POD_CONGESTED
- PodMemoryCongested
- POD_PENDING_REQUEST_CONGESTED
- PodMemoryDoC
- POD_CPU_CONGESTED

Maintain

Error logs are generated when the system is congested and the actions taken to bring the system back to normal. Warning logs are generated to indicate the congestion level. However, error logs are not generated when messages are rejected to avoid additional resource usage to write error logs.

If you encounter alerts at system or application levels, see Alerts section for resolution steps.

In case the alerts still persist, perform the following:

- Collect the logs: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

4.26 Support for UDR Discovery Using Group ID

PCF interfaces with UDR to receive subscriber related data for the UE (uePolicySet) AM (amPolicyData) and SM (smPolicyData) information. On PCF receiving requests from AMF or SMF, it performs UDR discovery using NRF.

With user equipment existing in millions, thousands of UDR discovery requests are sent to NRF. Many UE's subscription data will be associated to a single UDR in form of UDR groups. PCF, by performing UDR discovery using the UDR group id reduces the number of discovery requests to NRF for the same UDR. The **UDR Group ID** refers to one or more UDR instances managing a specific set of SUPI's.

SMF/AMF makes following PCF `GET` and initial `POST` requests with UDR group id `oc-policy-udr-group-id-list` to receive Policy associations associated with the user or SUPI.

- AMF for AM Policy association, POST ../npcf-am-policy-control/v./policies/ (PolicyAssociationRequest)
- AMF for UE Policy association,POST ../npcf-ue-policy-control/v./policies/ (PolicyAssociationRequest)
- SMF for SM Policy association,POST ../npcf-smpolicycontrol/v./sm-policies (SmPolicyContextData)

Core services (SM/ AM/UE) accepts requests containing `oc-policy-udr-group-id-list` header from NF consumer and forwards it to PDS. PDS forwards it to UDR connector. Based on the discovery parameters and the NF communication profile configurations received in the request, UDR connector selects query parameters and forms a request toward NRF client for UDR discovery or rediscovery.

For example the request toward NRF:

```
GET ../nnrf-disc/v./nf-instances?target-nf-type=UDR&requester-nf-type=PCF&group-id-list=<Value from XXX-UDR-Group-Id>&data-set=POLICY
```

Based on query parameters for UDR discovery or rediscovery and configurations, NRF client supports either non-SUPI or SUPI based caching.

Customizing Header Name Containing udr-group-id-list

PCF receives the service requests with header that contain UDR Group Id associated with the user/SUPI. Default custom header name is `oc-policy-udr-group-id-list`. The user shall be able to customize the header name as per their requirements at `routesConfig` in PCF custom_values.yaml file. Adding header for the following routes:

- SMF for SM Policy association - `sm_create_session_route`
- AMF for UE Policy association - `ue_create_session_route`
- AMF for AM Policy association - `am_create_session_route`

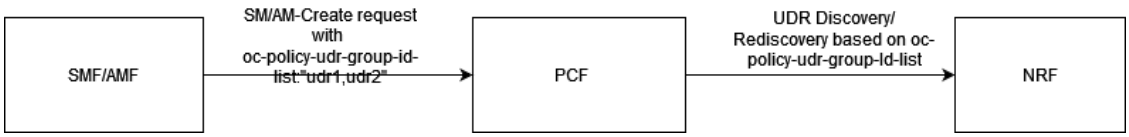
An example of default header structure in the custom_values.yaml file:

```
routesConfig:
    - id: sm_create_session_route
      uri: http://{{ .Release.Name }}-occnp-pcf-sm:
{{ .Values.global.servicePorts.pcfSmServiceHttp }}
      path: /npcf-smpolicycontrol/*/sm-policies
      order: 1
      method: POST
      readBodyForLog: true
      filters:
        subLog: true,CREATE,SM
        customReqHeaderEntryFilter:
          headers:
            - methods:
              - POST
              headersList:
                - headerName: 3gpp-Sbi-Message-Priority
                  defaultVal: 24
                  source: incomingReq
                  sourceHeader: 3gpp-Sbi-Message-Priority
                  override: false
                - headerName: oc-policy-udr-group-id-list
                  source: incomingReq
                  sourceHeader: oc-policy-udr-group-id-list
                  override: false
```

UDR Discovery in Model B and Model C Communications

SMF/AMF sends SM/AM Policy association requests with `oc-policy-udr-group-id-list` to PCF. And then PCF performs UDR discovery/rediscovery based on `oc-policy-udr-group-id-list` request header to NRF.

Figure 4-41 UDR Discovery with oc-policy-udr-group-id-list Header to NRF



During the GET and initial POST requests, PCF along with existing query parameters, `target-nf-set-id, data-set, preferred-locality, dnn, snssais, guami` for a NF communication profile, now will also have the following parameters for UDR discovery.

- group-id-list
- supi

The following table shows the query parameters (supi, group-id-list) that go from UDR connector to NRF Client for UDR discovery.

Table 4-16 Query Parameters

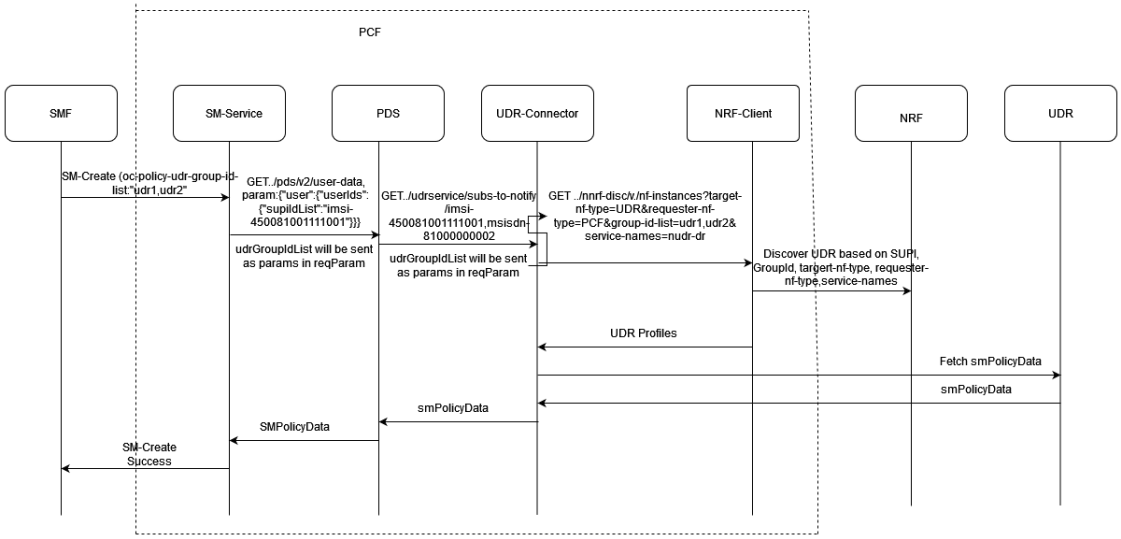
Query Parameters Selected to Sent from UDR Connector to NRF Client	Discovery Parameters Received by UDR Connector from PDS	Query Parameters Sent to NRF Client for UDR Discovery
group-id-list	group-id-list, supi	group-id-list
group-id-list	supi	supi
supi	group-id-list, supi	supi
supi	supi	supi
supi and group-id-list	group-id-list, supi	group-id-list, supi
supi and group-id-list	supi	supi
NF Communication Profile is not configured	supi	supi
NF Communication Profile is not configured	Group-Id-List, SUPI	SUPI

Note

- UE Policy uses the UDR selected for the UE during AM Policy association. When UDR_RELATED_RESOURCE for `amPolicyData` and `UEPolicySet` are enabled, PCF does not perform UDR discovery again during UE Policy association for the UE for which AM Policy association is already established
- UE Policy gets NFProfiles again from NRF if the same UDR that sent amPolicyData during AM Policy association is giving error and when the UDR_RELATED_RESOURCE for `amPolicyData` and `UEPolicySet` are enabled.

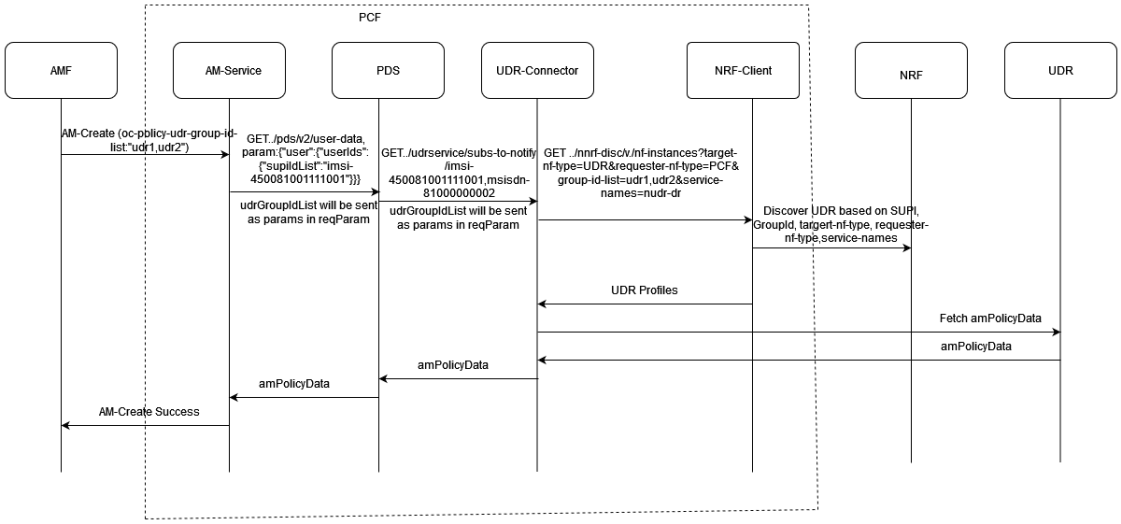
Call Flows

Figure 4-42 SMF sends SM Policy Association Request with oc-policy-udr-group-id



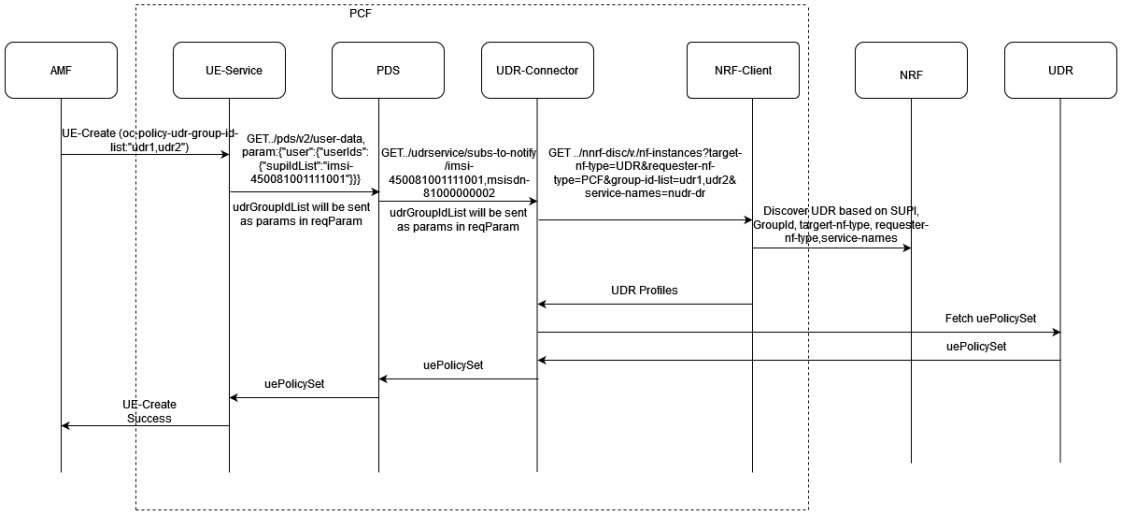
1. SMF sends SM Create request with `oc-policy-udr-group-id-list:"udr1,udr2"` to SM service.
2. SM service sends a GET request along with udr group id lists as request parameters to PDS.
3. PDS sends a GET request along with udr group id lists as request parameters to UDR Connector
4. UDR Connector sends a GET request along with udr group id lists as request parameters to NRF Client.
5. NRF Client sends UDR discovery request along with SUPI, GroupId, target-nf-type,requester-nf-type, and service-name request parameters to UDR via NRF.
6. NRF Client sends the UDR profiles to UDR Connector.
7. UDR responds back with `smpolicydata` to UDR Connector.
8. UDR connector forwards the response to SM service via PDS.
9. With successful creation of SM Policy association, SM service responds back to SMF with SM Create success.

Figure 4-43 AMF sends AM Policy Association Request with oc-policy-udr-group-id



1. AMF sends AM Create request with `oc-policy-udr-group-id-list:"udr1,udr2"` to AM service.
2. AM service sends a GET request along with udr group id lists as request parameters to PDS.
3. PDS sends a GET request along with udr group id lists as request parameters to UDR Connector
4. UDR Connector sends a GET request along with udr group id lists as request parameters to NRF Client.
5. NRF Client sends UDR discovery request along with SUPI, GroupId, target-nf-type,requester-nf-type, and service-name request parameters to UDR via NRF.
6. NRF Client sends the UDR profiles to UDR Connector.
7. UDR responds back with `ampolicydata` to UDR Connector.
8. UDR connector forwards the response to AM service via PDS.
9. With successful creation of AM Policy association, AM service responds back to AMF with AM Create success.

Figure 4-44 AMF sends UE Policy Association Request with oc-policy-udr-group-id



1. AMF sends UE Create request with `oc-policy-udr-group-id-list:"udr1,udr2"` to UE service.
2. UE service sends a GET request along with udr group id lists as request parameters to PDS.
3. PDS sends a GET request along with udr group id lists as request parameters to UDR Connector
4. UDR Connector sends a GET request along with udr group id lists as request parameters to NRF Client.
5. NRF Client sends UDR discovery request along with SUPI, GroupId, target-nf-type,requester-nf-type, and service-name request parameters to UDR via NRF.
6. NRF Client sends the UDR profiles to UDR Connector.
7. UDR responds back with `uePolicySet` to UDR Connector.
8. UDR connector forwards the response to UE service via PDS.
9. With successful creation of UE Policy association, UE service responds back to AMF with UE Create success.

Policy Features Impacted due to UDR Discovery using Group-id-list

Impact on Non-SUPI based Caching: When non-SUPI based caching is enabled, UDR discovery/rediscovery is done using `group-id-list`. NRF client cache's the discovered NF profiles along with query parameters. The following table shows if NRF Client cache's the NF profile or not when non-SUPI based caching is enabled:

Table 4-17 NRF Client Caching

NRF Agent Query Parameters Configurations	Query Parameter for UDR Discovery at NRF Client	Caching at NRF Client
group-id-list	group-id-list	True
group-id-list	group-id-list, supi	False
group-id-list	supi	False

Table 4-17 (Cont.) NRF Client Caching

NRF Agent Query Parameters Configurations	Query Parameter for UDR Discovery at NRF Client	Caching at NRF Client
Other than group-id-list	group-id-list	False (as query parameters sent to NRF client and configured should be same)

For more information about non-supi based caching, see the Support for Non-SUPI based On-Demand Discovery Caching of NF Profiles section.

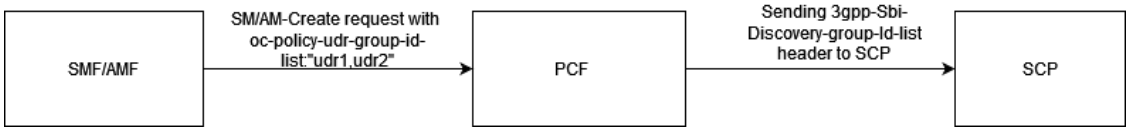
Impact on SUPI-based Caching: When SUPI-based caching is enabled, UDR discovery/rediscovery is done using `SUPI` and `(SUPI,Group-Id-List)`. Enabling the non-SUPI-based caching will not be applicable as the query parameter contains SUPI. For more information, see Support for SUPI based NRF Discovery Optimization and Response Caching from UDR section.

Impact on Session Retry: For `GET` and initial `POST` requests, this feature is not impacted and the UDR discovery happens based on query parameters as configured in NF communication profile. For more information, see Support for Session Retry and Alternate Route Service section.

UDR Discovery in Model D Communication

SMF/AMF sends SM/AM Policy association requests with `oc-policy-udr-group-id-list` to PCF. And then PCF performs UDR discovery using `3gpp-Sbi-Discovery-group-Id-list` request header to SCP.

Figure 4-45 UDR Discovery with 3gpp-Sbi-Discovery-group-Id-list Header to SCP



UDR connector compares NF communication profile configurations, NF discovery settings, Model D configurations and discovery parameters received from PDS and accordingly sends the request header to SCP based on the following table:

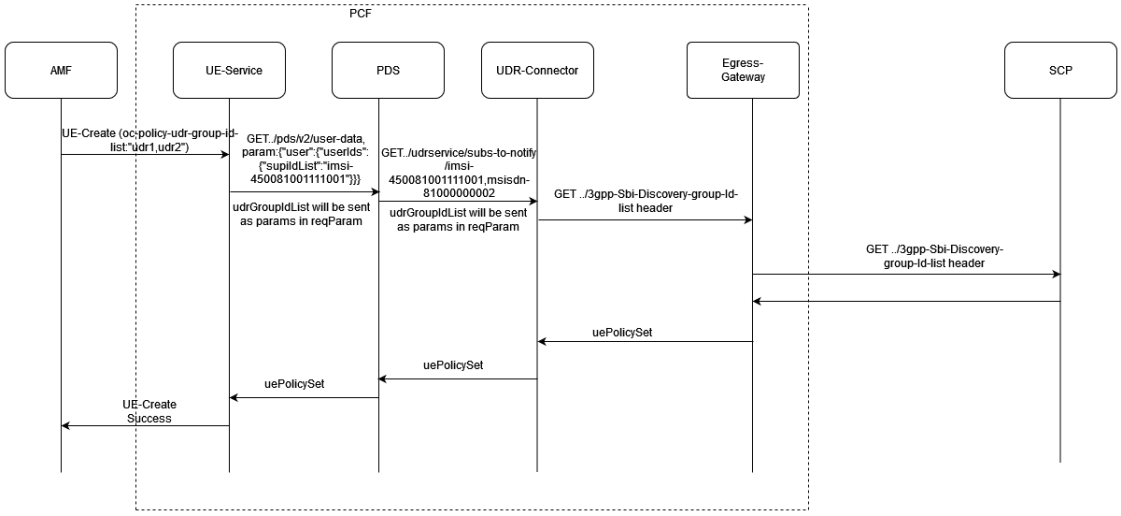
Table 4-18 Query Parameters

NF Communication Configurations	Values Received by UDR Connector	Header Sent to SCP
udr-group-id-list	udr-group-id-list, supi	3gpp-Sbi-Discovery-group-Id-list
udr-group-id-list	supi	No Header
supi	udr-group-id-list, supi	3gpp-Sbi-Discovery-SUPI
supi	supi	3gpp-Sbi-Discovery-SUPI
udr-group-id-list, supi	udr-group-id-list, supi	3gpp-Sbi-Discovery-group-Id-list, 3gpp-Sbi-Discovery-SUPI
udr-group-Id-list, supi	supi	3gpp-Sbi-Discovery-SUPI

> **Note**
>
> If configured, the UDR discovery happens for subsequent sessions for the same subscriber based on udr-group-id-list.

Call Flow

Figure 4-46 Model D, AMF sends UE Policy Association Request with oc-policy-udr-group-id-list



1. AMF sends UE Create request with `oc-policy-udr-group-id-list:"udr1,udr2"` to UE service.
2. UE service sends a GET request along with udr group id lists as request parameters to PDS.
3. PDS sends a GET request along with udr group id lists as request parameters to UDR Connector
4. UDR Connector sends a GET request along with 3gpp-Sbi-Discovery-group-Id-list header to Egress Gateway.
5. Egress Gateways sends a GET request along with 3gpp-Sbi-Discovery-group-Id-list header to SCP.
6. SCP responds with `uePolicySet` to Egress Gateway.
7. Egress Gateways forwards `uePolicySet` to UDR Connector.
8. UDR connector forwards the response to UE service via PDS.
9. With successful creation of UE Policy association, UE service responds back to AMF with UE Create success.

Managing Supports UDR Discovery using Group ID

Enable

This forms Policy applications core feature functionality. You do not need to enable or disable this feature.

Helm

You can customize the custom headers using `headerName` and `sourceHeader` routing parameters for SM/UE/AM service in `custom_values.yaml` file. For more information about routing configurations, see the "Custom Header Name for UDR Group Id" section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.

Configure Using CNC Console

The initial discovery parameters for Model B and C Communication profiles now supports the following parameters:

- `target-nf-set-id`
- `supi`
- `group-id-list`

Model D Communication profile supports the following parameters:

- `target-nf-set-id`
- `supi`
- `data-set`
- `preferred-locality`
- `dnn`
- `snssais`
- `guami`
- `group-id-list`

For more information about these discovery parameters, see the NF Communication Profiles section in the **Common Data** group, under **Service** settings in CNC Console.

The on demand discovery of caching in NRF client supports a new query parmeter `group-id-list`. For more information, see the NRF Agent section under **Service** settings in CNC Console.

Maintain
If you encounter alerts at system or application levels, see Alerts section for resolution steps.

In case the alerts still persist, perform the following:

- Collect the logs: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

4.27 Network Policies

Network Policies are an application-centric construct that allows you to specify how a pod communicates with various network entities. It creates pod-level rules to control communication between the cluster's pods and services, and to determine which pods and services can access one another inside a cluster.

Previously, the pods under Policy deployment could be contacted by any other pods in the Kubernetes cluster without any restrictions. Now, Network Policies provide namespace-level isolation, which allows secured communications to and from Policy with rules defined in

respective Network Policies. The Network Policies enforce access restrictions for all the applicable data flows except communication from Kubernetes node to pod for invoking container probe. For example, PCF internal microservices cannot be contacted directly by any other pods.

The key purpose of implementing Network Policies is to support principle of zero trust, the concept that no service or network can be trusted, including in house services and networks.

The following table lists the different access policies to be used by Policy traffic flows.

> **Note**
>
> This list is not exhaustive but tries to represent all the data flows supported by Policy traffic flows.

Microservice	Direction	Client/Server	Port	Access Policy
Configuration Svc	Egress	• Database • K8s API server for K8s secret	3306, K8s API Server Port	K8s Network Policies
Configuration Svc	Egress	• Jaeger Agent	6831	K8s Network Policies
Configuration Svc	Ingress	• Console • Egress Gateway for configuration • Ingress Gateway for configuration • Perf-info for configuration • App-info for configuration • ATS • ARS • NrfClient	8081	K8s Network Policies
Configuration Svc	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
Ingress Gateway	Egress	• Jaeger Agent	6831	K8s Network Policies
Ingress Gateway	Egress	• Database • K8s API Server for K8s Secret	3306, K8s API Server Port	K8s Network Policies
Ingress Gateway	Egress	• Coherence	8000, 7	K8s Network Policies
Ingress Gateway	Ingress	• Perf Info	8000	K8s Network Policies
Ingress Gateway	Ingress	• SBI Peer	80, 443	3GPP-defined Access Policies
Ingress Gateway	Ingress	• Coherence	7, 8000, 8095, 8096	K8s Network Policies

Microservice	Direction	Client/Server	Port	Access Policy
Ingress Gateway	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
Egress Gateway	Egress	• Jaeger Agent	6831	No Access Policy due to SBI Egress*
Egress Gateway	Egress	• Database • K8s API Server for K8s Secret	3306	No Access Policy due to SBI Egress*
Egress Gateway	Egress	• Coherence	7, 8000	No Access Policy due to SBI Egress*
Egress Gateway	Egress	• SBI Peer	Decided at run-time	3GPP-defined Access Policies
Egress Gateway	Egress	• ARS	ARS Port	K8s Network Policies
Egress Gateway	Ingress	• Egress Gateway for coherence	7, 8000	K8s Network Policies
Egress Gateway	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
Egress Gateway	Ingress	• Coherence	7, 8000	K8s Network Policies
Audit	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
Audit	Egress	• Database • K8s API Server for K8s Secret	3306, K8s API Server Port	K8s Network Policies
App Info	Ingress	• Registration • Subscription • Auditor	8000	K8s Network Policies
App Info	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
App Info	Egress	• DB monitoring port	8080	K8s Network Policies
diam-gateway	Egress	• Jaeger Agent	6831	K8s Network Policies
diam-gateway	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
diam-gateway	Egress	• Database • K8s API Server for K8s Secret	3306, K8s API Server Port	K8s Network Policies
diam-gateway	Ingress	• Peer	3868	K8s Network Policies

Microservice	Direction	Client/Server	Port	Access Policy
Query-Svc	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
Query-Svc	Egress	• Jaeger Agent	6831	K8s Network Policies
User svc	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
User svc	Ingress	• Monitoring	8000, 9443	K8s Network Policies
User svc	Egress	• Jaeger	6831	K8s Network Policies
Policy DS	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
Policy DS	Ingress	• Monitoring	8000, 9443	K8s Network Policies
Policy DS	Egress	• Database • K8s API Server for K8s Secret	3306, K8s API Server Port	K8s Network Policies
Policy DS	Egress	• Jaeger	6831	K8s Network Policies
Notifier	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
Notifier	Egress	• Jaeger	6831	K8s Network Policies
Notifier	Egress	• Signaling	8080	K8s Network Policies
Nwdaf-agent	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
Nwdaf-agent	Egress	• Database • K8s API Server for K8s Secret	3306, K8s API Server Port	K8s Network Policies
Nwdaf-agent	Egress	• Jaeger	6831	K8s Network Policies
Bulwark	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
Bulwark	Egress	• Signaling	8080	K8s Network Policies
Bulwark	Egress	• Jaeger	6831	K8s Network Policies
Performance	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies

Microservice	Direction	Client/Server	Port	Access Policy
Performance	Egress	• Signaling	8080	K8s Network Policies
alternate route svc	Ingress	• Coherence	7, 8000, 8095, 8096	K8s Network Policies
alternate route svc	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
soap-connector	Egress	• Signaling	8000, 8443	K8s Network Policies
soap-connector	Egress	• Jaeger Agent	6831	K8s Network Policies
soap-connector	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
soap-connector	Ingress	• Monitoring	9000	K8s Network Policies
usage-mon	Egress	• Jaeger Agent	6831	K8s Network Policies
usage-mon	Egress	• Database • K8s API Server for K8s Secret	3306	K8s Network Policies
usage-mon	Egress	• Signaling	8000	K8s Network Policies
usage-mon	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
ldap-gateway	Egress	• Jaeger Agent	6831	K8s Network Policies
ldap-gateway	Egress	• Signaling	8000, 8443	K8s Network Policies
ldap-gateway	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
diam-connector	Egress	• Signaling	8000	K8s Network Policies
diam-connector	Ingress	• Diameter	3868	K8s Network Policies
diam-connector	Ingress	• Monitoring	9000	K8s Network Policies
diam-connector	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
Biding	Egress	• Signaling	8000, 9443	K8s Network Policies
Biding	Egress	• Jaeger Agent	6831	K8s Network Policies

Microservice	Direction	Client/Server	Port	Access Policy
Biding	Egress	• Database • K8s API Server for K8s Secret	3306, K8s API Server Port	K8s Network Policies
Biding	Ingress	• Monitoring	9000	K8s Network Policies
Biding	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
pcrf-core	Egress	• Signaling	8000, 9443	K8s Network Policies
pcrf-core	Egress	• Jaeger Agent	6831	K8s Network Policies
pcrf-core	Ingress	• Diameter	3868	K8s Network Policies
pcrf-core	Ingress	• Monitoring	9000	K8s Network Policies
pcrf-core	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
SM Service	Egress	• Jaeger Agent	6831	K8s Network Policies
SM Service	Egress	• Signaling	8000, 9443	K8s Network Policies
SM Service	Egress	• Database • K8s API Server for K8s Secret	3306, K8s API Server Port	K8s Network Policies
SM Service	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
SM Service	Ingress	• Monitoring	9000	K8s Network Policies
AM Service	Egress	• Jaeger Agent	6831	K8s Network Policies
AM Service	Egress	• Database • K8s API Server for K8s Secret	3306, K8s API Server Port	K8s Network Policies
AM Service	Egress	• Signaling	8000, 9443	K8s Network Policies
AM Service	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
AM Service	Ingress	• Monitoring	9000	K8s Network Policies
UE Service	Egress	• Jaeger Agent	6831	K8s Network Policies

Microservice	Direction	Client/Server	Port	Access Policy
UE Service	Egress	• Database • K8s API Server for K8s Secret	3306, K8s API Server Port	K8s Network Policies
UE Service	Egress	• Signaling	8000, 9443	K8s Network Policies
UE Service	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
UE Service	Ingress	• Monitoring	9000	K8s Network Policies
PRE	Egress	• Jaeger Agent	6831	K8s Network Policies
PRE	Egress	• Signaling	8000	K8s Network Policies
PRE	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies
PRE	Ingress	• Monitoring	9000	K8s Network Policies
nrf-client	Egress	• Jaeger Agent	6831	K8s Network Policies
nrf-client	Egress	• Performance • Platform	8000	
nrf-client	Egress	• Cache Service	8095, 8096	
nrf-client	Egress	• Database • K8s API Server for K8s Secret	3306, K8s API Server Port	K8s Network Policies
nrf-client	Ingress	• Prometheus • Liveness • Readiness	9000	K8s Network Policies

As an assumption when deploying Policy, the following labels are set by default:

Table 4-19 Default Labels

Pod	Label
<name_of_the_pod>	np **Note**: This label is only used as long as NRF pods do not use the label app.kubernetes.io/part-of with occnp value.
<name_of_each_service>	app.kubernetes.io/name:

`np` is custom global label that must be added as follows:

1. Open `occnp-custom-values.yaml` file.
2. Add the label under `customExtention-allResources-labels` section.

For example:

```
customExtension:
   # The `factoryLabelTemplates` and `factoryAnnotationTemplates` can
   # accept templates rather than plain text.
   factoryLabelTemplates: {}
   factoryAnnotationTemplates: {}
   allResources:
     labels: {
       "np": "cnpolicy"
     }
```

3. Run the following Helm upgrade command.

```
helm upgrade <helm_name> <build_file.tgz> -f <custom_values.yaml> --
namespace
    <namespace>
```

When the upgrade is complete, the new tag must be observed on all pods.

PCF Security Policies:

- **deny-ingress-all**: To block all ingress traffic of pods present in a PCF deployment.
- **allow-ingress-network-sbi**: To allow traffic on the Ingress Gateway Pods on container ports 8000 and 9443 to allow sbi traffic.
- **allow-ingress-diameter-traffic**: To allow Ingress traffic for diameter gateway, diameter connnector, and PCRF Core on port 3868.
- **allow-ingress-local-prometheus**: To allow Ingress traffic for all services on port 9000.
- **allow-ingress-local-policy-flow**: To allow Ingress traffic between pods inside PCF.
- **allow-ingress-network-gui**: To allow ingress traffic on port 8000 for CM Service to access CNC Console.
- **deny-egress-all**: To block all egress traffic for all the pods.
- **allow-egress-network-all-gws**: To allow Egress traffic on all ports for these pods [occnp-egress-gateway, diam-gateway, ldap-gateway, occnp-alternate-route].
- **allow-egress-network-pre**: To allow Egress traffic on all ports for PRE (only use this when PER is enabled).
- **allow-egress-local-database**: To allow Egress traffic on ports 3306 & 8080 for connection with database.
- **allow-egress-local-k8sapi**: To allow Egress traffic on port 6443.
- **allow-egress-local-jaeger**: To allow Egress traffic on port 6831.
- **allow-egress-local-dns**: To allow Egress traffic on port 53.
- **allow-egress-local-flow**: To allow Egress traffic between pods inside PCF.

> **Note**
>
> The default Network Policies to be applied for Policy are the recommended even though they are not very granular but they keep operational overhead to the minimum and still achieve access control security.

Managing Network Policies

Enable

To use this feature, Network Policies need to be applied to the namespace wherein Policy is applied.

Configure

You can configure this feature using Helm. For information about configuring network policy for Policy deployment, see *Configuring Network Policy* section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.

Observe

There are no specific metrics and alerts required for the Network Policies feature.

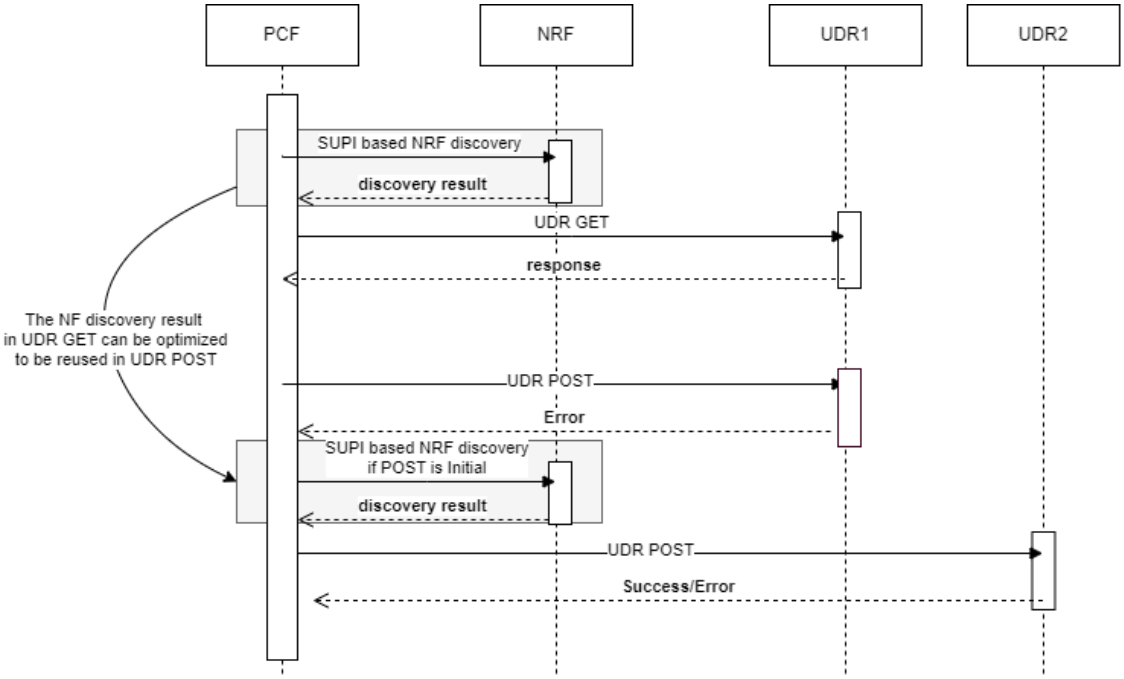
4.28 Support for SUPI based NRF Discovery Optimization and Response Caching from UDR

This feature helps in optimizing SUPI based NRF discoveries between the query and subscription (GET and POST) of the policy data request towards UDR.

PCF discovers the NF Profiles from NRF along with the query parameters used in discovery request, which are discovered on demand based on SUPI. SUPI based NRF discovery result for UDR GET request is cached till the POST request is completed. The on demand discovery based on SUPI is currently supported only for UDR discovery. The cache for the discovered UDR profiles expires even if the UDR POST result is success or fail.

The following diagram shows the SUPI based NRF discovery result for UDR GET request is cached till the POST request is completed.

Figure 4-47 UDR GET Discovery Result Reused in UDR POST

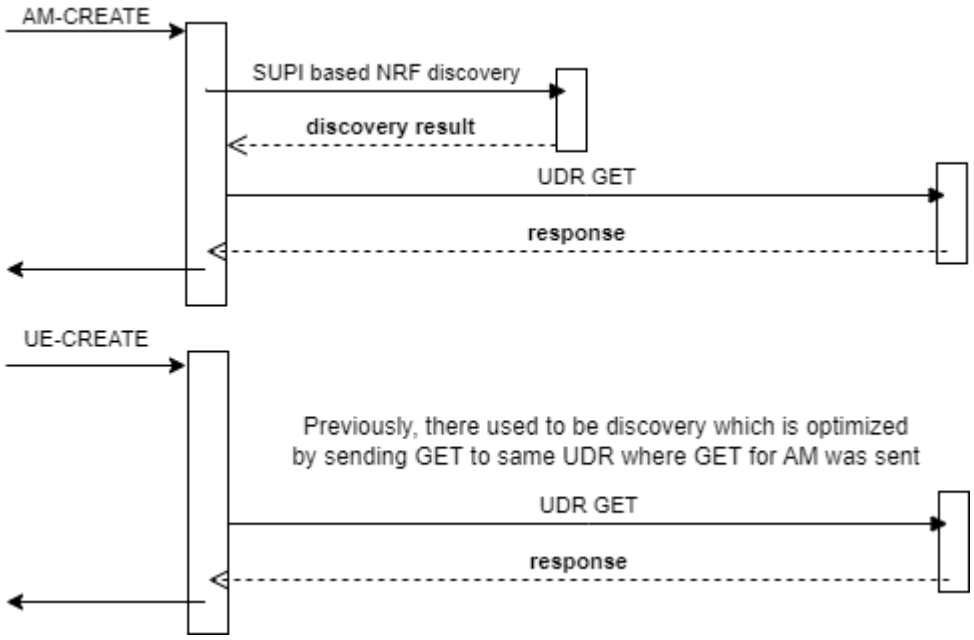


1. PCF sends SUPI based NRF discovery request to NRF.
2. NRF discovers the NF profile using SUPI.
3. PCF sends UDR GET request to UDR1.
4. UDR1 sends response message to PCF.
5. PCF sends UDR POST request to UDR1.
6. UDR 1 responds with an error.
7. PCF does SUPI based NRF discovery if the POST is initial.
8. NRF provides the corresponding discovery result.
9. PCF sends UDR POST to UDR2.
10. UDR2 responds with success or error. If it is successful, cache data is cleared. If it fails and there are no futher retry attempts, then POST fails and cache data is cleared.

Instead of doing discovery for amPolicyData and uePolicySet separately, the UDR information used in fetching amPolicyData can be reused in fetching uePolicySet and vice versa. PCF can rediscover the NF profiles for UDR based on the query parameter SUPI, from NRF during POST, if POST fails and uses the discovered profiles from the cache that was cached during GET.

The following diagram shows the reuse of UDR profile for amPolicyData GET in uePolicySet GET.

Figure 4-48 Reuse of UDR profile for amPolicyData GET in uePolicySet GET



1. For AM-Create request, PCF sends SUPI based NRF discovery request to NRF.
2. NRF discovers the NF profile using SUPI.
3. PCF sends UDR GET request to UDR1.
4. UDR1 sends response message to PCF.
5. For UE-Create request, PCF sends the GET to same UDR where GET for AM was sent.
6. UDR1 sends corresponding response message to PCF.

Managing the Feature

This section explains the procedure to enable and configure the feature.

Enable and Configure

By default, this feature is disabled. You can enable using CNC Console by configuring the following advanced settings:

- Set value of the following keys under the Advanced Settings section on the PDS Settings page:
  - UDR_FETCH_RELATED_RESOURCE
  - UPDATE_SSV_DATASOURCE_INFO_ON_CHANGE

  For information about how to configure for in CNC Console, see PDS Settings.
- Set value of the following keys under the Advanced Settings section on the PCF User Connector page:
  - UDR_NF_PROFILE_COOKIE_ENABLED
  - UDR_NF_PROFILE_COOKIE_COMPRESS
  - UDR_NF_PROFILE_COOKIE_LIMIT
  - UDR_GET_USE_RELATED_RESOURCE

For information about how to configure for in CNC Console, see PCF User Connector.

Observe

The following metrics have been added for this feature:

- Policy DS Metrics
  - occnp_nf_cookie_forwarded_total
  - occnp_nf_cookie_recieved_total

  For more information, see Policy DS Metrics.
- User Service Metrics
  - occnp_udr_nf_cookie_enabled_total
  - occnp_udr_use_related_resource

  For more information, see User Service Metrics .

4.29 RAA Error Code Handling

RAA error code handling helps in error handling at SM and PA services. The operator should be able to take corrective actions on the receipt of error response from Diameter Connector for initiated update-notify or RAR message.

The operator can configure the corrective action on the error handling configuration page for PA service. Based on this configuration, the PA service receives the error response from Diameter Connector. The Diameter Connector forwards the received diameter error result code in the error header of the response.

Following are the actions that PA service can use:

- Terminate the transaction: To terminate the transaction or call.

  > **Note**
  >
  > If error handling configuration is not done, the default behavior is to terminate the transaction.

- Cleanup session: To delete the corresponding Rx session locally. The PA service initiates the delete request internally that causes the corresponding PCC rule to be removed by SM service by sending Update_Notify request to SMF and also send dependentContextBinding delete request. In this case, no more ASR or STR messages get exchanged.

  > **Note**
  >
  > In case of 5002 DIAMETER_UNKNOWN_SESSION_ID, the default behavior is to cleanup the session. The 'Terminate the transaction' action should be configured only in case of an explicit requirement.

> **Note**
>
> There is no retry action of update notify http request (RAR) by error handling library at SM/PA service and all the attempts for retry for RAR must be performed as part of diameter session retry at Diameter Gateway.

Managing RAA Error Code Handling

This section explains the procedure to enable and configure the feature.

Enable

To enable RAA error handling, set the value of SYSTEM.PA_ERROR_HANDLER_ENABLED key to true in the Advanced Settings of PA service. For more information, see PCF Policy Authorization.

Configure

Once the service is enabled, you can configure the RAA error handling functionality under **Error Handling** on CNC Console for Policy. For information about how to configure for in CNC Console, see Error Handling.

Observe

The following metrics have been added in PCF Session Management service for this feature:

- error_handler_exec_total
- error_handler_in_total
- error_handler_out_total

For more information, see SM Service Metrics.

4.30 IPv6 Support in Converged Policy Mode

CNC Policy and its microservices can be deployed in a K8s Kubernetes environment that supports the IPv6 addressing. It supports the complete range of IPv6 prefixes as defined in the 3GPP TS 29.571 and 3GPP TS 29.510 specifications.

> **Note**
>
> CNC Policy supports either IPv6 or IPv4 protocol for communication. Dual IPs support is currently not available.

The protocol configurations can be performed during deployment using the Custom Values YAML file for CNC Policy. For more information about the confgurations, see *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*.

External Communications

With the support for IPv6 protocol, CNC Policy allows communication with external NFs over the following interfaces:

- The DNS (Domain Name System) for the resolution of FQDNs

- The SBI interface over the HTTP/2 and HTTPS protocols for the following types of connections:
  - Consumer Connections
  - Producer Connections
- Diameter interface for the following types of connections:
  - Connection as a server. For example, connection with PGW
  - Connection as a client. For example, connection to OCS over Sy
- Service IP endpoints
- Provisioning interface for connection with Cloud Native Core Console (CNCC)
- CNC Policy Rest API interface for the REST API Client configurations over IPv6

Internal Communications

CnPolicy supports all the internal communication over IPv6 addresses for PCF, PCRF, and Converged Policy modes.

The internal communications include:

- Inter microservice communication
- Communication with CnDbTier
- Communication with IGW, EGW
- Communication with Diameter Gateway and Diameter connector
- DNS for FQDN resolution
- Integration with common services using IPv6 addressing for observability - logging, monitoring and tracing
- Communication with platform components supporting retrieval of certificates/public/private keys

Managing IPv6 Support in Converged Policy Mode

Deploy and Configure

To deploy Policy in the IPv6 environment:

1. The cnDBTier must have IPv6 enabled. For information about installing cnDBTier, see "Installing cnDBTier" in Oracle Communications cnDBTier Installation Guide.
2. Deploy Policy over the IPv6 supported cnDBTier site. For information about installing and deploying Policy, see Oracle Communications Converged Policy Installation and Upgrade Guide.

You need to configure the IPv6 support during installation using the Custom Values YAML file for CNC Policy. The following parameters must be updated in the custom values file for CNC Policy:

Table 4-20 IPv6 Parameters

Parameter	Description
ingress-gateway.isIpv6Enabled	Set the value to true for this parameter when Policy is deployed in IPv6 cluster.

Table 4-20 (Cont.) IPv6 Parameters

Parameter	Description
egress-gateway.isIpv6Enabled	Set the value to true for this parameter when Policy is deployed in IPv6 cluster.
alternate-route.isIpv6Enabled	Set the value to true for this parameter when Policy is deployed in IPv6 cluster.
diam-gateway.envSupportedIpAddressType	Set the value to IPV6 for this parameter when Policy is deployed in IPv6 cluster.
global.isIpvSixSetup	Set the value to "true" if you are going to require HTTP communication over IPv6.

For more information about configuring the parameter value, see "Customizing CNC Policy" in *Oracle Communications Converged Policy Installation and Upgrade Guide*.

4.31 Handling Rx Stale Sessions

Policy Control Function (PCF) supports auditing the Rx session periodically to detect and remove stale sessions. This helps to avoid unlimited system memory utilization growth.

An Rx session is considered as stale when an association exists in PCF, but is not used by Proxy Call Session Control Function (P-CSCF) or Application Function (AF) for a specific time period. The time period after which the session is considered as stale is configurable.

PCF detects an Rx stale session based on:

- Configured Time To Live (TTL)
- Authorization Lifetime using:
  - Support of Rx Subscription Expiry
  - Authorization Lifetime Attribute Value Pair (AVP)

TTL is the time period for which an Rx session can be active. When the TTL of an Rx session reaches its maximum as configured by the user, the Rx session will be considered as stale.

Authorization Lifetime is the maximum lifetime of an Rx session. When the life time of a session reaches its Authorization Life Time, the session is considered as stale and is cleaned up.

> **Note**
>
> Currently, the value of Authorization Lifetime is just echoed back and it is expected to be the same value as configured in TTL under Audit settings in Policy Authorization service (PA service).

The P-CSCF provides Authorization Lifetime AVP and the `Support of Rx Subscription Expiry` feature bit for the Supported-Feature AVP in Rx Authorization Authentication Request Initial (AAR-I).

PCF echoes back the value in Authorization Lifetime AVP in Authorization Authentication Answer (AAA) response to P-CSCF along with the `Support of Rx Subscription Expiry` feature bit in the Supported-Feature AVP.

The P-CSCF sends Authorization Authentication Request Update (AAR-U) according to the Authorization Lifetime AVP.

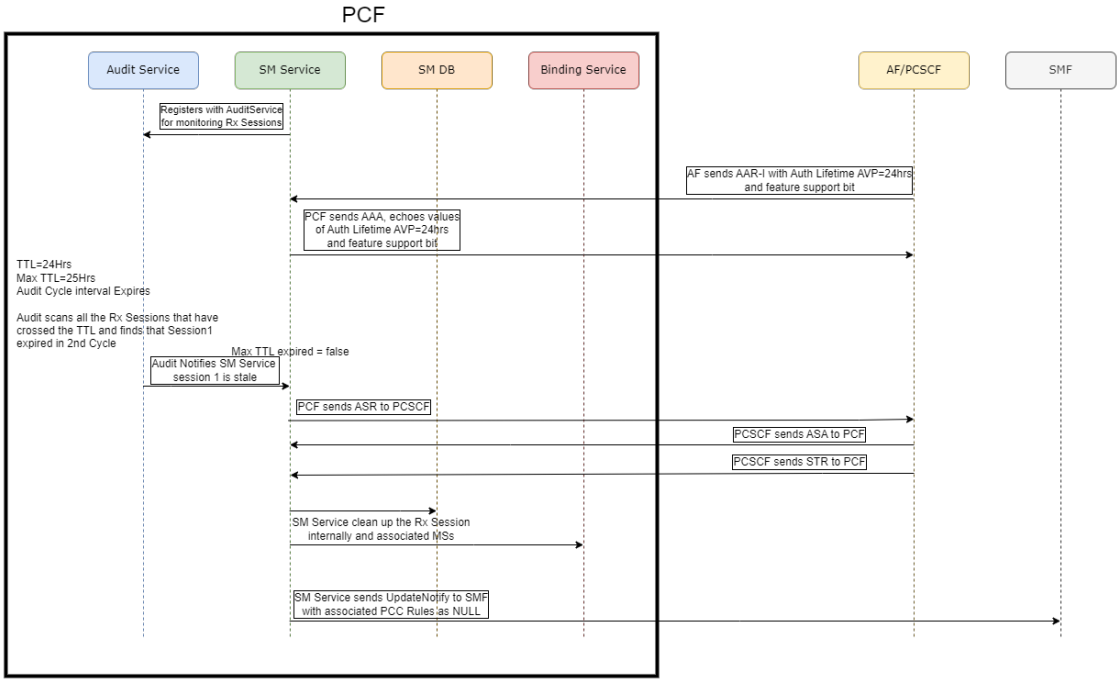
PCF updates the last access timestamp of the Rx session when it receives AAR-U from P-CSCF.

PCF sends abort session request to P-CSCF and removes the stale Rx Session locally along with the related PCC rules in SMF, upon detection.

Detection and removal of stale Rx session generates appropriate alert and log entries.

Call Flow

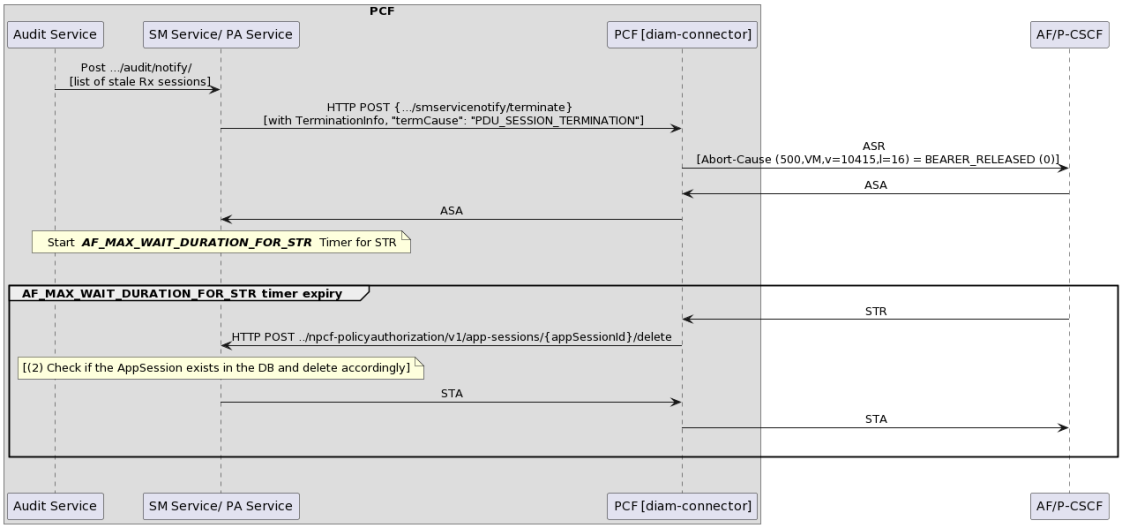
Figure 4-49 Rx Stale Session Detection and Deletion Based on TTL and Authorization-Lifetime



1. If the **Enable Audit** field is enabled under **Audit** group in **PCF Policy Authorization** page for **Service Configurations**, SM service registers with Audit service for monitoring Rx sessions.
2. SM service receives AAR-I message from P-CSCF with `Support of Rx Subscription Expiry` feature bit for the Supported-Feature AVP.
3. If the **Enable Authorization Lifetime** field is enabled under **System** group in **PCF Policy Authorization** page for **Service Configurations**, and the AAR-I message includes Authorization-Lifetime AVP, SM service sends Authorization-Lifetime to P-CSCF in Rx AAA messages. It returns the same value of Authorization-Lifetime AVP to P-CSCF, as the value received from P-CSCF.
4. SM service registers with the Audit service for monitoring the Rx session.
5. Audit service analyzes the EXPIRY_TIMESTAMP and compares it with TTL value. If necessary, it sends a message to SM service as mentioned in the next step.

6. If AAR-U for an Rx session is not received and its TTL expires, Audit service sends a message to SM service with `Max TTL Expired = true`.
7. SM service sends an Abort Session Request (ASR) to P-CSCF for that Rx session.
8. P-CSCF sends a successful Abort Session Answer (ASA) response to SM service.
9. P-CSCF also sends a STR to SM service.
10. SM service responds to P-CSCF with STA message.

Figure 4-50 Rx session cleanup when Audit service detects a stale session



If **Enable Audit** field is enabled, SM service registers with Audit service for auditing Rx sessions. Depending on the value of `EXPIRY_TIMESTAMP` field in `AppSession` database, the Audit service considers that the Max TTL of an Rx session has expired. SM service updates `EXPIRY_TIMESTAMP` after every successful AAR-I and AAR-U messages. In case of georedundancy, the Audit service uses `SITEID` field in `AppSession` database to filter and handle the Rx sessions for multiple sites.

1. When Audit service detects that some of the Rx sessions expired and for any reason these Rx sessions are not deleted after TTL expiry, Audit service sends the list of such Rx sessions to PA service for which the Max TTL has expired.
2. For all the Rx sessions in the list, PA service sends PDU Session Termination request to SM service.
3. SM service sends ASR to P-CSCF and waits for a configured time period to receive a successful Abort Session Answer (ASA) response from P-CSCF.
    - When sending ASR, if SM service receives any response that is not success (except for timeout), it cleans up the session.
        - If ASA times out, SM service checks if AppSession exists for these sessions in Audit service. SM service deletes the Rx Session and the associated resources internally. Also, it deletes the associated PCC Rules from SMF externally, by sending Update Notification to SMF.
        - If SM service receives ASA with error code other than 5002 (for reasons like the ASR request did not go out of PCF), then the Audit service does not delete the AppSession for such sessions. Depending on the configuration, SM service resends the ASR to P-CSCF. If all the retries fail with timeout or any other error,

SM service deletes the Rx Session and associated resources internally. Also, it deletes the associated PCC Rules from SMF externally, by sending Update Notification to SMF.

- If SM service receives a successful ASA from P-CSCF, it waits for receiving a STR from P-CSCF.

4. PCF uses `AF_MAX_WAIT_DURATION_FOR_STR` timer to receive the STR from P-CSCF.

   If `AF_MAX_WAIT_DURATION_FOR_STR` timer expires and SM service does not receive STR from P-CSCF, SM service checks if AppSession exists for these sessions. SM service deletes the Rx Session and the associated resources internally. Also, it deletes the associated PCC Rules from SMF externally, by sending Update Notification to SMF.

5. If SM service receives a successful STR from P-CSCF, SM service checks if AppSession exists for these sessions. SM service cleans up these Rx sessions and the associated resources externally and internally. Also, it deletes the associated PCC Rules from SMF externally, by sending Update Notification to SMF.
6. SM service responds to P-CSCF with STA.

- When there are bulk of stale Rx records detected, in order to avoid performance impact on the system, Policy allows to configure:
  - the rate at which ASR is sent to P-CSCF.
  - the rate at which the records are deleted.

Managing Handling Rx Stale Sessions

Enable

By default, this feature is disabled. You can enable stale Rx session audit and deletion using CNC Console or REST API for Policy.

Enable using CNC Console:

To enable the Rx Stale Session detection based on TTL and Authorization-Lifetime:

- To enable auditing by Audit service, configure:
  - **Enable Audit** parameter to `true` under **Audit** group in **PCF Policy Authorization** page for **Service Configurations**.
  - Configure **App Session Age (in minutes)** parameter under **Audit** group in **PCF Policy Authorization** page for **Service Configurations**.
- To enable detection and deletion of stale sessions based on Authorization Lifetime, enable **Enable Authorization Lifetime** parameter under **Systems** group in **PCF Policy Authorization** page for **Service Configurations**.

For more information about enabling the feature through CNC Console, see PCF Policy Authorization.

Enable using REST API

To enable the feature, set the value of `enable` parameter to `true` in `Audit` group in `Policy Authorization Service` API.

For more information about enabling the feature through REST API, see *Policy Authorization Service* in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

You can configure this feature using the CNC Console or REST API for Policy.

Configure using CNC Console: Perform the feature configurations under **Audit** group in **PCF Policy Authorization** page for **Service Configurations**. For more information, see PCF Policy Authorization.

Configure using REST API: Configure the parameters under `Audit` group in `Policy Authorization Service` API. For more information about configuring the feature through REST API, see *Policy Authorization Service* in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

> **Note**
>
> - On upgrade, `SITEID` and `EXPIRY_TIMESTAMP` contains null values in all AppSession records until there is an update on the AppSession.
> - As `SITEID` and `EXPIRY_TIMESTAMP` contains null values, the default value of `Handle NULL As Stale` parameter is set to `false`. You must enable this setting after the desired TTL has passed after site upgrade. This will cleanup all AppSessions for which `SITEID` and `EXPIRY_TIMESTAMP` contains null value. That is, they did not receive an update for the duration after upgrade and now these sessions are considered as stale.

Observability

Metrics

The following SM service metrics are used to differentiate between SM and PA deletions:

- audit_delete_records_max_ttl_count
- audit_notifications_sent
- audit_delete_records_count
- audit_terminate_notify

For more details on these metrics, see SM Service Metrics.

4.32 Data Compression

Data compression can be used to reduce the data size and hence the storage used by database. It can also be used to improve the communication latency between application and DB when compression is performed at the application level.

Following are the available options for data compression schemes:

- **MySQL_Compressed**: Compression and decompression performed at MySql level
- **Zlib_Compressed**: Compression and decompression performed at application level

Binding, PDS, and PCRF-core services support this feature only with Zlib_Compressed scheme. SM service supports this feature with both MySQL_Compressed and Zlib_Compressed schemes.

Managing the Feature

Enable

This feature is disabled by default. It can be enabled by choosing the compression scheme from respective service configuration. For more information, see the details of the **Data Compression Scheme** field in PDS, Binding Service, and PCF Session Management in the Configuring Policy Using CNC Console section.

Configure
You can configure this feature using Helm for PCF Session Management and CNC Console for PCF Session Management, PDS, and Binding service as follows:

- **Helm**: Set the value of the parameter `smDataCompressionScheme` and `paDataCompressionScheme` to 0, 1, or 2 during install or upgrade. For more information, see the Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide.
- **CNC Console**: You can configure this feature using the **Data Compression Scheme** field in PDS, Binding Servie, and PCF Session Management in the Configuring Policy Using CNC Console section.

Observe
The following metric has been added in PCF Session Management service for this feature:

- occnp_db_overall_processing_time

For more information, see SM Service Metrics.

4.33 Usage Monitoring on Gx Interface

Cloud Native 4G Policy implements Usage Monitoring or Quota Management using the combination of PCRF Core and Usage Monitoring. These two microservices have their respective policy engines and thus separate policy projects. Each of the two microservices (along with their respective policies) has the following functions with respect to Quota Management:

- PCRF Core - Controls PCC and/or Session Rules, and Charging Description.
- Usage Monitoring - Controls the Quota Selection, Accumulation and Grant.

For details on PCRF Core, see PCRF Core Service

For details on Usage Monitoring Service, see Usage Monitoring Service

To achieve end to end use cases of 4G quota, the policies must be split into these two microservice functions.

The configuration of Quota plans can be done at two places:

- CNC Policy - Quota plans that are not per subscriber can be configured directly on CNC Policy and used in Policy Configuration. Fore more details, see Usage Monitoring Service.
- UDR - Quota plans that need to be assigned per subscriber can be provisioned on the UDR.
  For details on UDR Quota plans, see *Converged Quota Support* section in *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

> **Note**
>
> Quota plans when configured in CNC Policy will avoid duplication of the plan data in UDR for every subscriber.

Quota Management for a Subscriber

Usage Monitoring supports different quota plans for a subscriber such as Basic, Dynamic, Daily, Roaming, Monthly quota based on volume of usage.

Basic Quota
Basic quota specifies restrictions on the amount of data volume, active session time, or service-specific events that a subscriber can consume.

A single quota can express limits on any combination of volume, time, or events. A Basic quota is also referred as a plan. For example, Domestic/Home/ Roaming plan.

Plan
A **plan** describes a subscriber's basic, recurring service. A plan can be a basic plan, or a pass or a top-up.

The quota plan for a subscriber is provisioned at UDR. For details on provisioning quota and dynamic quota for a subscriber, see *Convered Quota Support* section in *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

Plans include policy characteristics such as time and volume limits. These characteristics can be computed automatically or through policy rules. Policy actions grant plans, based on a subscriber's tier provisioned on CnUDR.

The following fields under **Usage Monitoring** page for **Service configurations** can be used to configure the time and volume limits for a plan:

- Minimum Volume Grant (bytes): Minimum grant value (in bytes) that can be approved for volume based usage monitoring.
- Maximum Volume Grant (bytes): Maximum grant value (in bytes) that can be approved for volume based usage monitoring.
- Minimum Time Grant (seconds): Minimum grant value (in seconds) that can be approved for time based usage monitoring.
- Maximum Time Grant (seconds): Maximum grant value (in seconds) that can be approved for time based usage

  monitoring.

The default volume and time grants for the plan can be configured using the fields available under **Default Volume Grant** and **Default Time Grant** sections in **Usage Monitoring** page.

A plan can have associated quota controls, which in turn can be subject to modification or over-ride through Passes, Top-ups, and Roll-overs.

Reset frequency can be:

- Hourly
- Daily
- Weekly
- Monthly
- Yearly
- Never

Postpaid monthly data plans are monthly recurring data plans that have defined **volume/time thresholds associated**. Users are notified about the user’s current usage at various levels.

On reaching the threshold, the user's access is subjected to suspension of service or reduced quality of service.

The default usage levels can be configured using the following fields under **Default Usage Levels** section in **Usage Monitoring** page for **Services Configuration** in CNC Console:

- Minor Usage Level (%): Threshold value in percentage indicating minor usage.
- Major Usage Level (%): Threshold value in percentage indicating major usage.
- Critical Usage Level (%): Threshold value in percentage indicating critical usage.
- Exhausted Usage Level (%): Threshold value in percentage indicating usage level has exhausted allowed limit.

Dynamic Quota
Dynamic quotas are one-time changes to a subscriber's service.

There are two types of dynamic quotas:

- Passes
- Top-ups.

Passes
Passes allow the service providers to grant resources outside of a subscriber's normal/default plan.

A pass is a one-time override quota that a subscriber can purchase that temporarily replaces or adds to a subscriber's default plan.

It allows users to access data till a predefined quota (either volume, time or event based) for a specific validity period.

For short time users, there can be different types of data passes:

- Hourly
- Daily
- Weekly
- Monthly
- Yearly
- Never

A subscriber can have multiple passes. An active Pass can modify the QoS controls, charging parameters, or other configurable rules associated with a subscriber's service.

A Pass may:

- be valid for a restricted interval
- start when provisioned, or at a specific time, or upon occurrence of a triggering event within its validity interval
- end at a specific time, or after given duration once activated, or upon a particular event
- apply continuously, or only during certain time periods, or only under certain conditions (e.g. when roaming)
- apply to the subscriber's overall usage, or be more limited (e.g. applying only to specific applications, flows, traffic types, or pre-defined rules)

Passes are common options for pre-paid subscribers, who frequently have limited or no data access via their basic plan, and may purchase Passes to gain access to such services. They

can also be used to allow Casual Use plans for pre- or post-paid subscribers to purchase services on an occasional basis which they would not otherwise subscribe for on an ongoing basis.

Users are notified about the user's current usage at various levels. On reaching the threshold, the user's access is subjected to suspension of service or reduced quality of service.

Pass configuration options can be modified. The dynamic PCC rules are updated based on the changes on the Pass Configuration.

Upon deletion of a pass, the pass is exhausted immediately.

Top-up
A Top-up allows to extend access to services beyond the time or volume limits typically enforced by a plan. A top-up is processed if the default plan is exhausted, or the quota is expired. There can be multiple Top-ups for a subscriber based on time or volume.

> **Note**
>
> Passes cannot not have Top-ups.

Top-up configuration options can be modified. The dynamic PCC rules are updated based on the changes on the Top-up configuration.

Upon deletion of a Top-up, the Top-up is exhausted immediately.

The quota plan for a subscriber is provisioned at UDR. For details on provisioning quota and dynamic quota for a subscriber, see *Convered Quota Support* section in *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

Quota details are configured at UDR based on ueID using the REST API: {apiRoot}/nudr-dr-prov/v1/policy-data/{ueId}. For more details on how to add the quota information, see *Provisioning Operations for PCF Data* section in *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide.*

Whenever there is a change in the subscriber quota information, UDR notifies Policy through updateNotify message.

The order/priority of basic plan, pass or top-ups can be specified using **Selection Order 1**, **Selection Order 2**, and **Selection Order 3** fields under **Data Limit Selection** section in **Usage Monitoring** page for **Service configurations** in CNC Console for Policy. For more information, see Usage Monitoring Service.

Exhaustion (Quota)
Exhaustion occurs when reports indicate that usage of a metered unit has equalled or exceeded the specified quota limit. If a recurring quota is exhausted, typically the subscriber's sessions are subjected to more restrictive policies until the end of the plan period or billing cycle.

Expiration (Quota)

Expiration occurs when a periodic plan reaches the end of the plan period or billing cycle, or when a one-time pass or top-up reaches its established End Time or close to the validity period.

> **Note**
>
> The time-based expiration of a Quota is quite different from the exhaustion of a Quota restricting the active session Time of a subscriber's usage.

A plan (periodic quota) is typically reset at expiration.

Rollover
A rollover allows a subscriber to carry forward unused units from one billing cycle to another.

For example, if a subscriber is allowed 10 GB of data a month and only uses 9 GB, the remaining 1 GB of data can be saved for use in the next month.

Rollover units can accumulate and can be carried across multiple months.

Data units are rolled over based on multiple conditions such as:

- Rollover unused data when it is more than a minimum threshold.
- Total data (base data + rollover data) may not exceed a maximum cap.
- Rollover certain percentage of unused data.
- Data may be rolled over for a fixed number of billing cycles.

When data is rolled over without any restriction, the leftover volume/time can be added to the base volume/time in the next billing cycle.

When data is rolled over with restrictions on duration of usage (for example, a maximum number of rollover periods/cycles), it needs to be tracked as separate rollover definitions (data limits).

The data rollover for a plan can be enabled using **Data Rollover** field under **Usage Monitoring** page for **Service Configurations**.

The data rollover can be configured based on a particular profile. The rollover profiles can be created by configuring the fields in **Data Rollover Profiles** page under **Usage Monitoring** in **Policy Data Configurations**.

You can create multiple profiles and specify profile to be used for the plan using **Data Rollover Profile** field under **Usage Monitoring** page for **Service Configurations**.

Also, you can specify the default rollover profile to be used in **Default Data Rollover Profile** field.

Support for Multiple Passes for a subscriber
Usage Monitoring supports One time passes.

Support for Multiple Top-Ups
Usage Monitoring supports One time top-ups.

Matchlist for Quota use cases
To identify the geographical location where the data usage is occurring as well as to ensure that the correct quota plan is applied, Policy uses match lists while defining the MCC/MNC allocations, RAT-Type, Service GW IP address, APN Lists.

- IPv4 Subnet - When a Match List is created with a data type of IPv4 subnet, the Items are configured with valid IPv4 subnet addresses in CIDR notation.

  For example, 192.168.10.0/24.

PRE evaluates if an IPv4 address belongs to one of the subnets configured in the match list.

- IPv6 Subnet - When a Match List is created with a data type of IPv6 subnet, the Items are configured with valid IPv6 subnet addresses in CIDR notation.

  For example, 2001:db8:abcd:0012::0/64.

  PRE evaluates if an IPv6 address or an IPv6 Prefix belongs to one of the subnets configured in the match list.

- Regular Expression - When a Match List is created with a data type of Regular Expression, the Items are configured with valid Regular Expressions, such as ^([a-zA-Z0-9. _-])+$.

  PRE evaluates if a string matches one of the Regular Expressions configured in the match list.

Support for flexible billing day change

Billing day for an active plan can be changed from monthly to yearly, weekly, daily, or hourly. The billing day change can be applied to either the current billing cycle or next billing cycle.

The **Apply Billing Day / Data Plan Change** field under **Usage Monitoring** page for **Service configurations** in CNC Console can be used to indicate whether the billing change should be applied to current billing cycle or to the next billing cycle.

With flexible billing day change, the billing day for plans will also be changed with the same periodicity. But, Policy supports changing the periodicity as well.

The billing date change should take place and determine if the usage should be pro-rated.

- Delayed behavior: Allows the billing day change to take effect on the next billing cycle irrespective of the current usage. Quota will be reset after the current billing cycle end date is reached.
- Immediate behavior: Allows the billing day change to take effect immediately. Quota will also be reset immediately.
- Default behavior: Allows the flexible billing day changes to take place immediately. For a user with no current usage and for a user that has current usage, billing day change will take place after the end of the user's current billing cycle is reached.

The **Enable Pro-rated Data at The Time of Billing Day Change** field under **Usage Monitoring** page for **Service configurations** can be used to indicate whether pro-rated data calculation should be applied to the selected plan at a time of billing day change.

Usage monitoring support at PCC rule level

Policy supports Usage monitoring either at PCC rule level or at session level.

Usage monitoring can be requested on an individual rule (PCC rule level) or on all rules activated during a IP-CAN session (session level).

- Based on certain custom attributes on CnUDR, quota can be granted for an application like whatsapp or facebook although the pass or a top-up volume been exhausted.
- Differentiate unlimited usage of certain application and not volume accumulate those applications for overall quota monitoring. For example: Plans with Facebook unlimited.

Multiple quotas contain single session-level and multiple PCC-level grants.

Rollover will be applicable for PCC-level grants.

- PCCRule data can be provisioned at UDR or configured at Policy. This can be of type BASE, TOP-UP and PASS.

- PCCRule can also be rollover. All rollover configurations will be applicable for PCCRule data limits.
- PCCRule can also allow excessUsage data.
- PCCRule can be of Type BASE, Top-up and pass.

You can configure the usage monitoring at PCC rule level using the following blocks available under **Policy Projects** in CNC Console.

- PCC Rule Hint: Allows to access the value of PCCRuleHint from Data limit profile and apply the same in UMPolicyDecision in monitoring key.
- Active Monitoring Key with PCCRuleHint: Allows to select a monitoring key for the configured PCCRule from usage monitoring policy decision which fulfills the value from active monitoring with PccRuleHint attribute.

For more information on configuring the PCC rules for usage monitoring, see *Usage Monitoring* section in *Oracle Communications Cloud Natvie Core, Policy Design Guide*.

Pro-Rating of quota limits

Policy supports pro-rating of quota volume. If a subscriber's profile billing day is changed, a shorter than usual period occurs (either from the date of change, or after the next scheduled reset).

For example, if the user has a plan with a monthly quota of 30GB and the plan start date is the 1st of every month, now the plan is activating at the 10th of that month, so rather than allocating the complete 30GB to the user, only 20GB will be allocated to the user as the expected activation date is the 1st of the month, but the plan activates after 10 days, so the remaining data is 20GB for that month.

The following parameters under **Usage Monitoring** page for **Service Configurations** can be used to configure the pro-rating of the quota:

- Enable Pro-rated Data at The Time of Activation: Iindicates whether pro-rated data calculation should be applied to the selected plan at a time of activation or not.
- Enable Pro-rated Data at The Time of Billing Day Change: Iindicates whether pro-rated data calculation should be applied to the selected plan at a time obilling day change .

Pro-rating can be configured per plan to reduce the quota allocation for the short period.

> **Note**
>
> Pro rating is not applicable for one-time plan.

Quota Policy conditions

The Policy supports a number of blocks, actions, and conditions that can be used for constructing Policy rules, such as:

- where the user has greater than number of rollover units of type unit type for plan name and usage type.
- where the user is using greater than 80 percent and less than 100 percent of total volume for Monthly1,Daily1
- where the user is using greater than 100 percent of total volume for Monthly1,Daily1

For more information on the blocks, actions, and conditions available to wrtite Policy rules, see see *Usage Monitoring* section in *Oracle Communications Cloud Natvie Core, Policy Design Guide*..

PRE support for Usage Monitoring

Policy uses Policy Runtime Engine (PRE) to enable and manage the following usage monitoring use cases:

- Grant pass or top-ups.
- Enable time of day quota. For example, If time of day is between HH:MM:SS - HH:MM:SS grant x% of volume and Y QoS
- Quota Plans can be activated by Policy Rule Conditions which look for attributes on
  - Gx attributes : MCC-MNC, RAT Type, Serving GW IP address.
  - CnUDR such as **Tier**, **Entitlement**, or **Custom Field(s)***values* provisioned in the Subscriber Profile

PRE performs the Policy evaluation when it receives a notification from UDR regarding quota or state data update.

Possible change of parameters include:

- Reset of Quota, triggers an action to throttle, block, or re-direct user to charging portal
- Billing day change triggers pro-rata calculation
- Subscriber Tier change
- Any customer attribute definition change

Support for profiles based configuration

Data Limit Profiles

When a subscriber is provisioned with multiple quota plans with different priorities, a separate profile can be used to track each plan. For example, if a subscriber is provisioned a daily plan and a monthly plan, one profile is used to track the daily quota limit and another profile is used to track the monthly quota limit.

A unique name and identifier can be assigned to each data limit profile and details such as plan type, profile type, parent plan, priority, usage level, reset period and billing day can be configured for the profile.

Data Limit Profiles can be configured using CNC Console and REST API.

- **Using CNC Console**: Apply the configurations in **Data Limit Profiles** page under **Usage Monitoring** in **Policy Data Configurations**.
  For more information, see Data Limit Profiles.
- **Using REST API**: Configure the parameters in UM Data Limit Profile API.
  For more information, see *UM Data Limit Profile* section in *Oracle Communications, Cloud Native Core, REST API Guide*.

Data Limit Selection Profiles

A set of rules can be used to select one or more data plans based on certain conditions.

For example, if a subscriber is provisioned with one monthly limit quota plan and two roaming plans: roam1 and roam2, a selection rule can be used to select a plan based on home zone and roaming zone. Separate selection profiles can be created for each selection rule.

A unique name can be assigned to each selection profile and details such as rule type, rule priority, parameter type (forwarded attribute/Policy decision tag), attribute name, and match list can be configured for the profile.

Data Limit Selection Profiles can be configured using CNC Console and REST API.

- **Using CNC Console**: Apply the configurations in **Data Limit Selection Profiles** page under **Usage Monitoring** in **Policy Data Configurations**.
  For more information, see Data Limit Selection Profiles.
- **Using REST API**: Configure the parameters in UM Data Limit Selection Profile API.
  For more information, see *UM Data Limit Selection Profile* section in *Oracle Communications, Cloud Native Core, REST API Guide*.

Data Limit Sorting Profiles

Sorting rule is a set of rules to sort data limits.

When a subscriber is provisioned with multiple quota plans, a sorting rule can be applied to sort the quota plan based on a criteria. For example, if a subscriber is provisioned with two Monthly Limit Quota plan with different priority plan1 - priority1 and plan2 - priority2, a sorting rule can be created to select plan with higher priority. Separate sorting profiles can be created for sorting rules.

The sorting rules can be configured based on order of index, parameter type, attribute name, and order.

If selection rules are configured, sorting rules are applied after applying selection rules and when the output of selection rules results in more than one data limit.

Data Limit Sorting Profiles can be configured using CNC Console and REST API.

- **Using CNC Console**: Apply the configurations in **Data Limit Sorting Profiles** page under **Usage Monitoring** in **Policy Data Configurations**.
  For more information, see Data Limit Sorting Profiles.
- **Using REST API**: Configure the parameters in UM Data Limit Sorting Profile API.
  For more information, see *Data Limit Sorting Profile* section in *Oracle Communications, Cloud Native Core, REST API Guide*.

Data Rollover Profiles

When data rollover is enabled, data rollover profile can be used to configure the rollover.

The data rollover profile can be configured with details such as:

- rollover data consumption
- maximum number of rollovers
- data rollover percentage
- maximum rollover threshold
- minimum rollover threshold
- data limit cap

Data Rollover Profiles can be configured using CNC Console and REST API.

- **Using CNC Console**: Apply the configurations in **Data Rollover Profiles** page under **Usage Monitoring** in **Policy Data Configurations**.
  For more information, see Data Rollover Profiles.
- **Using REST API**: Configure the parameters in UM Data Limit Sorting Profile API.

For more information, see *Data Rollover Profile* section in *Oracle Communications, Cloud Native Core, REST API Guide*.

Attribute Forwarding Profiles

Policy supports forwarding the value of attributes such as Serving Gateway MCC MNC / IP Address, and APN from PCRF Core to Usage Monitoring Service. An attribute forwarding profile is used to configure the list of attributes to be forwarded from PCRF Core to Usage Monitoring.

An attribute forwarding profile can be configured with details such as attribute name, attribute source, attribute selection (predefined/custom), request message type (diameter message/ HTTP message), and attribute location.

The attribute forwarding profile name must be specified in PCRF Core settings.

Attribute forwarding profiles can be configured using CNC Console or REST API:

- **Using CNC Console**: Apply the configurations in **Attribute Forwarding Profiles** page under **Common Data** for **Service Configurations**.
  For more information, see Attribute Forwarding Profiles.
- **Using REST API**: Configure the parameters in Attribute Forwarding Profile API.
  For more information, see *Attribute Forwarding Profile* section in *Oracle Communications, Cloud Native Core, REST API Guide*.

Data Compression in Usage Monitoring

In order to handle the growing size of the Usage Monitoring database and the replication volume, Policy supports compressing the Usage Monitoring data.

Usage Monitoring service application is used to reduce the V columns in `UmContext` table for data persisted in the database.

Data Compression Scheme flag available under **Usage Monitoring** page for **Service Configurations** in CNC Console is used to configure the compression status on the V columns in `UmContext` table.

When accessing the data from `UmContext.v`, the application determines the compression status based on the value of `COMPRESSION_SCHEME` and does the decompression accordingly.

The **Data Compression Scheme** flag accepts the following values:

- `Disabled`: Indicates that the data will not be compressed and will be stored in the column in uncompressed format. The value of `COMPRESSION_SCHEME` for that row is set to the default value 0.
- `MySQL Compressed`: Indicates that the data will be compressed using MySQL and will be stored in the column in compressed format. The value of `COMPRESSION_SCHEME` for that row is set to 1.
- `Application Compressed`: Indicates that the data will be compressed using Zlib and will be stored in the column in compressed format. The value of `COMPRESSION_SCHEME` for that row is set to 2.

> **Note**
>
> - The existing data will remain in uncompressed format and their COMPRESSION_SCHEME` is set to default NULL. Once the "Data Compression Scheme" changes to "MySQL Compressed" or "Application Compressed", all future v column data will be in compressed format.
> - Existing session's data will be compressed once we send CCR-U/UDR Notify with compression enabled.

For more information on configuring data compression for Usage Monitoring, see Configuring Usage Monitoring.

Managing the Feature

This section provides information on enabling, configuring, observability and logging for Usage Monitoring on Gx Interface.

Enable

By default, the Usage Monitoring service is disabled.

To enable the service, set the **Enabled** parameter under **Usage Monitoring** group in **PCRF Core Settings** page for **Service Configuration**. to `true` at the time of installing or upgrading Policy.

For information on how to set the parameter value, see *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*.

The **Enable PRE** field under **Usage Monitoring** page for **Service configurations** in CNC Console can be used to enable or disable the interaction with PRE for Policy evaluation as per the match list.

Configure

You can configure the Usage Monitoring for Gx interface using CNC Console and REST API.

- **Configure using CNC Console**: To configure Usage Monitoring for Gx interface using CNC Console, perform the configurations as described in the following sections:
  - Configuring Usage Monitoring
  - Settings
  - Attribute Forwarding Profiles
  - Usage Monitoring

  For details on the blocks used to create Policy projects, see *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.
- **Configure using REST API**: Refer to the *Usage Monitoring* section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

4.33.1 Migrating Subscribers from OCPM to PCRF Deployment

Quota details for both OCPM (4G) and Converged Policy (5G) are configured and provisioned at the individual subscriber level in OCUDR and CnUDR respectively.

For details on how CnUDR provisions and manages the quota information for PCRF, see *Convered Quota Support* section in *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

Using Provisioning Gateway, 4G provisioning system provisions OCPM data over SOAP/XML interface on CnUDR. Provisioning Gateway converts SOAP/XML to REST/JSON and forwards to CnUDR. CnUDR converges the OCPM quota and dynamic quota data to Converged data model and stores in CnUDR.

For details on how CnUDR integreates with the provisioning gateway, see *Cloud Native Core, Provisioning Gateway Interface Specification Guide*.

CnPCRF interacts with CnUDR to access and manage both OCPM and converged quota information. cnPCRF uses the N36 interface toward CnUDR for OCPM call flows.

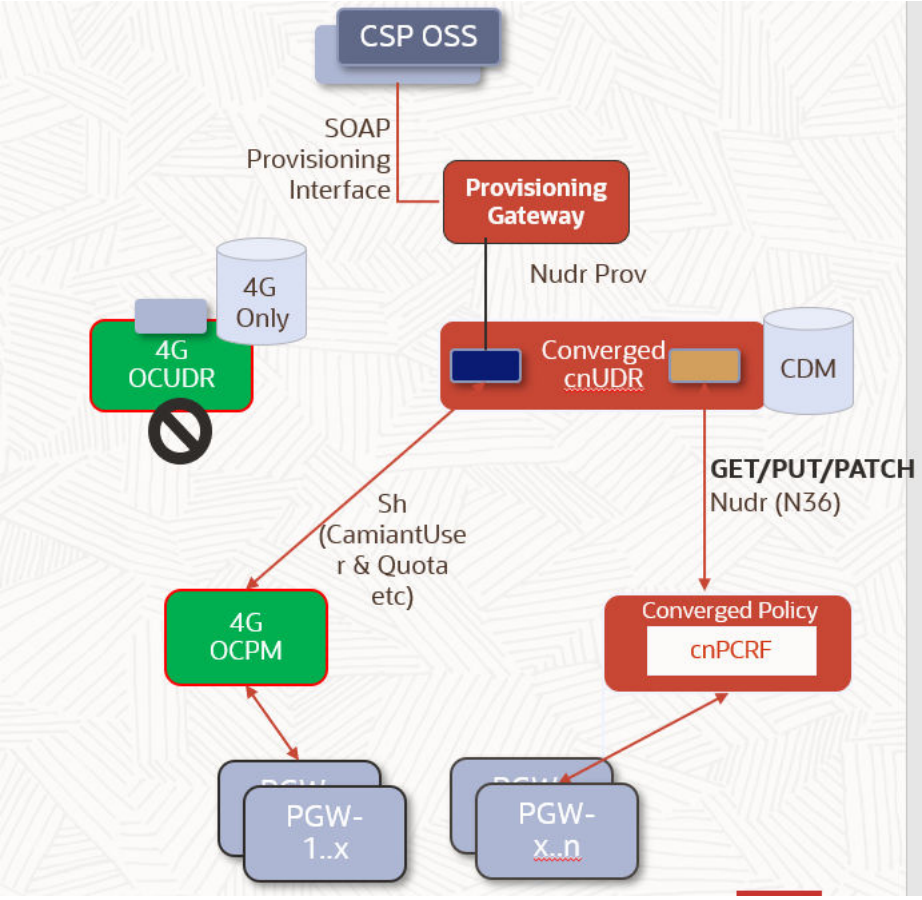
The OCPM data that CnPCRF receives from CnUDR includes:

- Quota data: the runtime quota data of a subscriber. That is, the data time and/or volume consumed by that subscriber and the other meta-data. For example, plan activation date, reset date, any roll-over plans and their corresponding usages.
  The remaining quota must be derived based on the consumed data retrieved from 4G UDR.
- Dynamic quota data: includes the provisioning data for dynamic plans which are also known as passes and top-ups.

Important

Since the 4G data structures for quota usages and monitoring are not defined by 3GPP standards, there must be a mapping from the proprietary 4G data structures to 3GPP defined 5G Data structures.

The following diagram depicts the connectivity between OCPM and CnUDR which will be swicthed to CnPCRF and CnUDR.



Diverting the subscribers from OCPM deployment to PCRF deployment involves the following procedures:

Migrating Subscribers from OCUDR to CnUDR

Subscribers from OCUDR are migrated to CnUDR either using the migration tool or on-demand.

Using the migration tool, the subscribers are migrated in batches based on subscriber range during the off peak hours.

The subscriber data is exported in EXML format, which is compatible with 4G OCUDR export format. It supports the export of 4G policy data (VSA and umData/umDataLimits) in EXML format from 5G UDR to 4G OCUDR using the subscriber export tool. In case of any error during the migration, the EXML format supports rollback of exported data.

For more details on the data export, see *Support for EXML Format* section in *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

> **Note**
>
> When Migration is in progress, dual provisioning at both OCUDR and CnUDR is not supported.

For more details on how the subscribers are migrated from OCUDR to CnUDR, see *Migrating Subscriber Data from 4G UDR to 5G UDR* section in *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

The migration of the subscriber data can be monitored using the metrics mentioned under *nudr-migration-service Metrics* section in *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

In case of migration failure for any reason either at OCUDR or at CnUDR, the quota for the subscriber must be reprovisioned to make up for any quota loss.

In case of new deployment, CnUDR must provision usage monitoring level, which indicates the level of the usage monitoring instance (PDU Session level or per Service). When `Enable Quota Migration` parameter is enabled, CnPCRF reads InitialServiceSpecific from OCUDR and sets the UsageMonLevel in CnUDR.

To set the UsageMonLevel, configure the following mandatory parameters in UDR for SM Policy data for subscriber profile:

Table 4-21 Mandatory Parameters

Parameter	Description
umDataLimits.endDate	Indicates the end date and time till when the usage monitoring level to be applied.
umDataLimits.name	Uniquely identifies the UM Data Limit Profile by a name.
umDataLimits.limitId	Specifies the limit identifier. Multiple limit identifiers may have the same priority.
umDataLimits.umLevel	Indicates the level of the usage monitoring instance (PDU Session level or per Service).
umDataLimits.startDate	Indicates the start date and time from when the usage monitoring level to be applied.
umDataLimits.usageLimit.duration	Indicates the duration of the usage limit in seconds. **Range**: 3600 - 31536000 (1 Year)
umDataLimits.usageLimit.totalVolume	Specifies the total data octets for both downlink and uplink. **Range**: 512 - 1073741824000 (1000 GB)
umDataLimits.usageLimit.uplinkVolume	Specifies the uplink data octets. **Range**: 512 - 1073741824000 (1000 GB)
umDataLimits.usageLimit.downlinkVolume	Specifies the downlink data octets. **Range**: 512 - 1073741824000 (1000 GB)
umDataLimits.resetPeriod.period	Indicates whether the periodicity is "YEARLY", "MONTHLY", "WEEKLY", "DAILY" or "HOURLY". **Default value**: Monthly
umDataLimits.Type	Indicates which day of the period to reset the usage limit. **Default value**: LAST_DAY **Note**: If this configuration is updated, the new configuration shall be applied in the next billing cycle.

Table 4-21 (Cont.) Mandatory Parameters

Parameter	Description
umDataLimits.Priority	A Priority is used to weigh it with other Data Limit Profiles and UDR provisioned Data Limits of the same Plan Type. The priority can be used by Policy or Data Limit Selection Profile for choosing among different candidate profiles. The Plan Type, Priority within that Plan Type and the Selection Order of Plan Types together decide the order of selection of Data Plans.

For disaster recovery purposes on CnUDR, the database is backed up as described in *Manual Database Backup and Restore* section in *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

Diverting the Sh Traffic from OCPM to CnUDR

OCPM interacts with CnUDR over Sh interface.

To migrate the incoming traffic from OCPM to CnUDR:

1. Log in to the OCPM Console with the given credentials.
2. Add the new data source.
   a. Under **Configuration** section for MPE_CLUSTER, click **Data Sources** tab.
   b. Add the new data source as explained in *Adding a Data Source* section in *Oracle Communications Policy Management Configuration Management Platform Wireless User's Guide*.
3. Configure Sh data source.
   a. Under **Configuration** section for MPE_CLUSTER, click **Data Sources** tab.
   b. Click **Modify**.
   c. Add the Sh data source as as explained in *Configuring an Sh Data Source* section in *Oracle Communications Policy Management Configuration Management Platform Wireless User's Guide*.
4. Configure the server information as explained in *Configuring Sh Server Information* section in *Oracle Communications Policy Management Configuration Management Platform Wireless User's Guide*. Perform all the necessary
5. **Reports** tab under **Configuration** section for MPE Clusters can be used to monitor the status and activity of the connection. For more details, see *Policy Server Reports* section in *Oracle Communications Policy Management Configuration Management Platform Wireless User's Guide*.

> **Note**
>
> If CnUDR receives any Sh-UDR request from OCPM and the subscriber profile is not yet migrated to CnUDR, CnUDR fetches the data from OCUDR on-demand and responds to OCPM.

On the CnUDR front, you can monitor the connectivity to OCPM using the metrics mentioned under *Diameter Gateway Metrics* and *nudr-diameterproxy-service Metrics* sections in *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

In case of errors from cnUDR, rollback the Sh connection from CnUDR to OCUDR by configuring the data source back to OCUDR and changing the server connectivity as explained in *Configuring Sh Server Information* section in *Oracle Communications Policy Management Configuration Management Platform Wireless User's Guide*.

Configuring CnPCRF to fetch the data from CnUDR

1. Configure the CnPCRF to fetch the data from CnUDR.
   a. Perform the following congurations in **Usage Monitoring** page under **Service Configurations** in CNC Console:
      i. Enable **Enable Quota Migration** field to allow retrieval of OCPM data from CnUDR.
      ii. Configure the following fields under **Custom Attribute Mapping**:
         - Reset Day & Time
         - Data Plan Name
         - Data Plan Priority
         - Data Plan Type
         - Data Rollover
         - Data Rollover Profile
         - Parent Plan Name
         - Parent Plan Source

         Data migrated from CnUDR includes the volume usage as is in custom attributes. Based on the Total available volume, CnPCRF calculates the InputVolume and OutputVolume for both Monthly quota and Dynamic Quota for Passes and Top-Ups.
      iii. Verify the priority of the plans and configure **Reverse Priority** field accordingly to make sure that the order or priority in both OCPM and CnPCRF are aligned.

      For more details, see *Configuring Usage Monitoring* section in *Oracle Communications Cloud Native Core, Converged Policy User Guide*.
   b. If Data Limit Sorting Profiles is enabled, configure the fields in **Data Limit Sorting Profiles** page under **Usage Monitoring** in **Policy Data Configurations**.
      For more details, see *Data Limit Sorting Profiles* section in *Oracle Communications Cloud Native Core, Converged Policy User Guide*.
   c. Create the Policy projects for OCPM subscribers using the blocks in **Policy Projects** under **Policy Management** in CNC Console.
      With more number of subscribers migrated from OCUDR to CnUDR, more number of subscribers can be configured on CnPCRF to fetch the profiles from CnUDR.

      For more details on writing polices, see *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.
   d. Once all the OCPM subscribers are migrated to CnPCRF, terminate the sessions at OCPM and configure the subscriber range to divert the subscribers to CnPCRF. Only CnPCRF will send the GET request to CnUDR for subscriber profiles.
   e. In case of any failure either at CnUDR or at CnPCRF during the migration:

i. Disable `Enable Quota Migration` flag

ii. Bring up the OCPM again.

iii. Configure the OCPM to fetch the data from CnUDR over Sh interface.

Any quota loss during the fallback from CnPCRF to OCPM is made up by reprovisioning the quota.

For details on the metrics used to monitor the successful migration at CnPCRF, see *Usage Monitoring Metrics* section in *Oracle Communications Cloud Native Core, Converged Policy User Guide*.

For detail on the metrics used to monitor the successful migration at CnUDR, see the following sections in *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*:

- nudr-dr-service Metrics
- nudr-notify-service Metrics
- nudr-ondemand-migration-service Metrics

> **Note**
>
> Configuring the PGW to connect to CnPCRF instead of OCPM and also setting the subscriber range at PGW is not in the scope of this document as PGW is outside the Oracle system.

4.34 Support for Autoenrollment of Subscribers

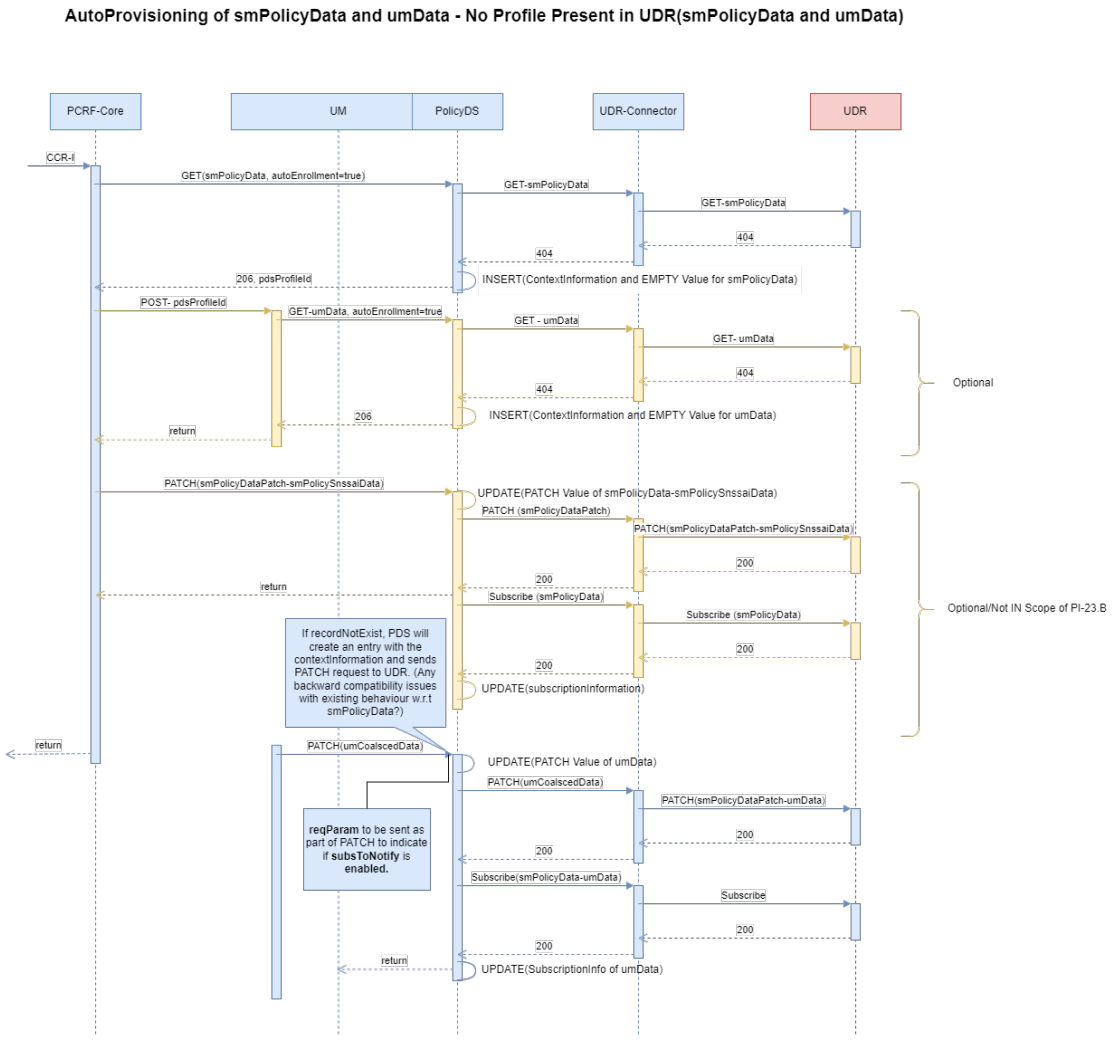
Policy supports autoprovisioning of the subscribers at UDR when the subscriber profile is not present in UDR.

1. When PCRF Core receives a CCR-i request, it tries to retrieve the subscriber profile from UDR.
2. If the subscriber profile is not present at UDR, PCRF Core handles the subscriber plan as configured at PCRF Core. The plan is chosen based on MCC-MNC, APN received in the request Message. It applies the right quota to the user and triggers a PATCH towards UDR.
3. A PATCH request is sent to UDR on SM-Data with details of policy-data including UM-Data.
4. UDR triggers autoprovisioning of the profile.

Call Flow

The following call flow depicts the scenario when the subscriber profile is not provisioned at UDR.

Figure 4-51 Scenario when user profile is not avaiable in UDR

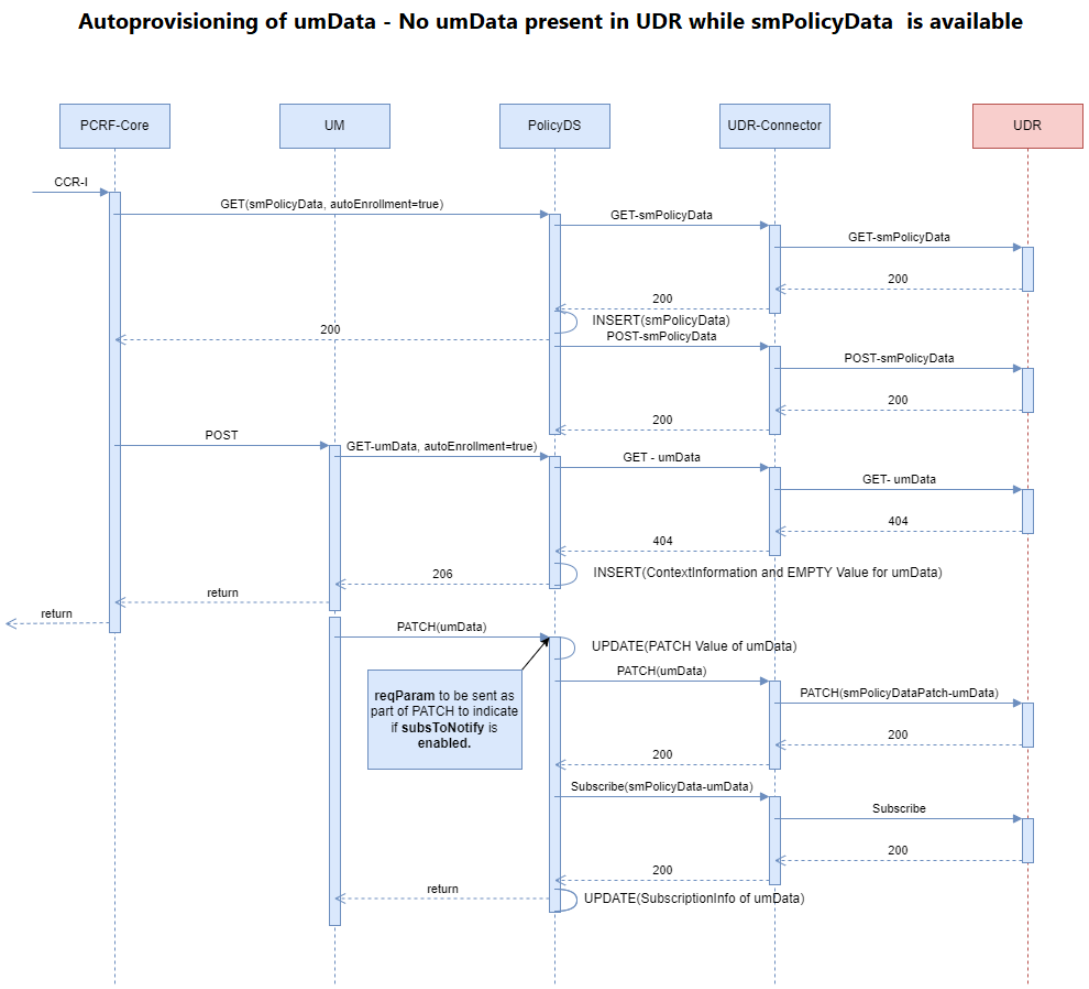


1. PCRF Core receives a CCR-I request.
2. PCRF Core sends a GET request to PDS with the autoenrollment flag set to *true*.
3. PDS sends the GET request to UDR Connector, which in turn forwards the request to UDR.
4. If the subscriber profile is not present at UDR, UDR responds with a 404 message.
5. UDR connector forwards the error message to PDS.
6. PDS receives the 404 response and then reads the autoenrollment flag. It generates the default SM Policy data and adds the last error code as 404 indicating that UDR sent the 404 error in the same SM Policy data.
7. PDS sends the generated SM Policy data to the core service.
8. Optionally, Usage Monitoring service can send a GET request to PDS requesting for the quota profile.
9. PDS sends the GET request to UDR Connector, which in turn forwards the request to UDR.

10. If the quota profile is not present at UDR, UDR responds with 404 error and UDR connector forwards the 404 error message to PDS.
11. PDS reads the autoenrollment flag and generates a default UMquota with last error code as 404 and sends the data to Usage Monitoring service.
12. The default UM data that is generated at PDS is inserted into the association table with all the context information.
13. The Usage Monitoring service generates quota based on APN, MCC and MNC and sends a PATCH request to PDS.
14. PDS creates an entry in the database and sends this PATCH request to UDR Connector, which in turn forwards the request to UDR.
15. Once the autoprovisioning is done at UDR, it responds with a 200 success message.
16. After the PATCH request is successful, PDS sends a subscription request to UDR to subscribe for umData.

The following call flow depicts the scenario when the subscriber profile is provisioned at UDR, but the umData is not provisioned.

Figure 4-52 Scenario when the subscriber profile is provisioned at UDR, but the umData is not provisioned



1. PCRF Core receives a CCR-I request.
2. PCRF Core sends a GET request to PDS with the autoenrollment flag set to *true*.
3. PDS sends the GET request to UDR Connector, which in turn forwards the request to UDR.
4. If the subscriber profile is present at UDR, UDR responds with a 200 ok message with the subscriber data.
5. UDR connector forwards the message to PDS.
6. PDS sends the SM Policy data to the core service.
7. Optionally, Usage Monitoring service can send a GET request to PDS requesting for the quota profile.
8. PDS sends the GET request to UDR Connector, which in turn forwards the request to UDR.
9. If the quota profile is not present at UDR, UDR responds with 404 error and UDR connector forwards the 404 error message to PDS.
10. PDS reads the autoenrollment flag and generates a default UMquota with last error code as 404 and sends the data to Usage Monitoring service.
11. The default UM data that is generated at PDS is inserted into the association table with all the context information.
12. The Usage Monitoring service generates a quota based on APN, NCC and MNC and sends a PATCH request to PDS.
13. PDS creates an entry in its database and sends this PATCH request to UDR Connector, which in turn forwards the request to UDR.
14. Once the autoenrollment is done at UDR, it responds with a 200 success message.
15. After the PATCH request is successful, PDS sends a subscription request to UDR to subscribe for umData.

Managing the Feature

This section provides information on enabling, configuring, observability and logging for Support for autoenrollment for the subscriber.

Enable

You can enable the support for autoenrollment of the subscribers using CNC Console.

To enable the autoprovisioning of the subscriber profile at UDR when the subscriber profile is not available, enable `Auto Enrollment on UDR` parameter for `SmPolicyData` under **User** group in **PCRF Core Settings** page for **Service Configuration**.

For more information, see Settings.

Configure

You can configure the the support for autoenrollment of the subscribers using CNC Console.

For more information, see *SmPolicyData* under **User** group in Settings section.

4.35 Support for Subscriber Notification Using SMPP

Policy supports sending subscriber notification (SMS alerts regarding Quota management or usage reporting and activation/deactivation of services) to subscribers through SMPP (Short

Message Peer-to-Peer) protocol. This feature enables transfer of short messages between External Short Message Entities (ESME) and Message Centres (MC).

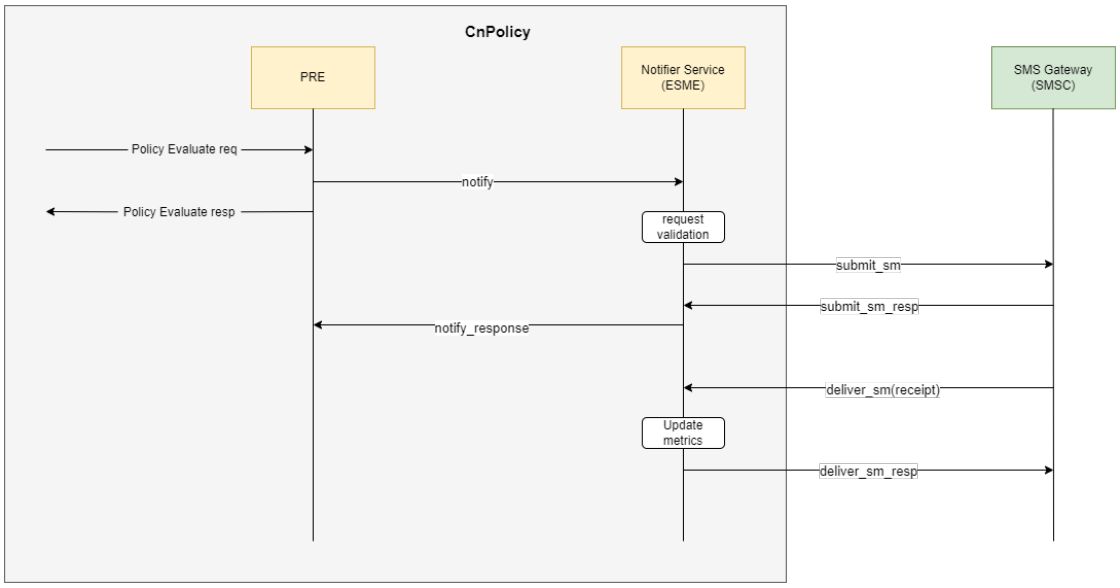
To support notification through SMPP, Policy uses the HTTP notification framework.

Policy supports this feature using Notifier Service. You must enable the Notifier Service to use this feature. For more information, see Notifier Service.

> **Note**
>
> Currently, this feature is supported only for PCRF-Core call flows.

Call Flow

Figure 4-53 Call flow for sending messages using SMPP protocol



1. PRE receives a Policy Evaluate request.
2. After the evaluation, PRE sends a notification message to the Notifier Service.
3. Notifier service evaluates the request and sends the notification to SMS Gateway (SMSC).
4. The SMS Gateway acknowledges the notification request.
5. The SMS Gateway sends a delivery receipt to Notifier Service.
6. The Notifier Service acknowledges the delivery receipt to the SMS Gateway.
7. The Notifier Service updates the metrics and then notify response to PRE.
8. PRE sends the Policy evaluation response to the core service.

Managing Support for Subscriber Notification Using SMPP

Enable

User can enable Subscriber Notification through SMPP using CNC Console or REST API for Policy.

- **Enable using CNC Console**: Set the value of **Enable SMPP** parameter to `true` under **Notifier Configurations** page in CNC Console.
  For more information, see Notifier Configurations.
- **Enable using REST API**: Set the value of **smppConfiguration.isEnabled** parameter under **{apiRoot}/oc-cnpolicy-configuration/v1/notifier** API to `true`.
  For more information, see *Notifier Configurations* section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

User can configure Subscriber Notification through SMPP using CNC Console or REST API for Policy.

- **Configure using CNC Console**: To configure the Subscriber Notification through SMPP, perform the configurations under **Notifier Configurations**, **Notification Server**, **SMS Gateway Group**, and **SMSC Host Info** pages in CNC Console.
  For more information, see Notifier Configurations.
- **Configure using REST API**: To configure the Subscriber Notification through SMPP use the following REST APIs:
  - To enable and configure the notification through SMPP: **{apiRoot}/oc-cnpolicy-configuration/v1/notifier**
  - To configure the SMS Gateway: **{apiRoot}/oc-cnpolicy-configuration/v1/smsGateway**
  - To configure the SMSC Host Info: **{apiRoot}/oc-cnpolicy-configuration/v1/smsHostInfo**

  For more information, see *Notifier Configurations* section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Observe

The following metrics are added to support the subscriber notification using SMPP:

- `http_in_conn_request_total`
- `http_in_conn_response_total`
- `smpp_request_total`
- `smpp_response_total`
- `active_smsc_conn_count`

For more details on metrics, see Notifier Metrics.

The `SMSC_CONNECTION_DOWN` alert is added to trigger an alarm when the connection to Short Message Service Center (SMSC) Host is down.

For more information, see SMSC_CONNECTION_DOWN.

4.36 Support for Query on Update and Subscription to UDR

PCF uses the Query data procedure to retreive user data from the UDR, when Policy core services such as SM, AM, or UE receives an update request.

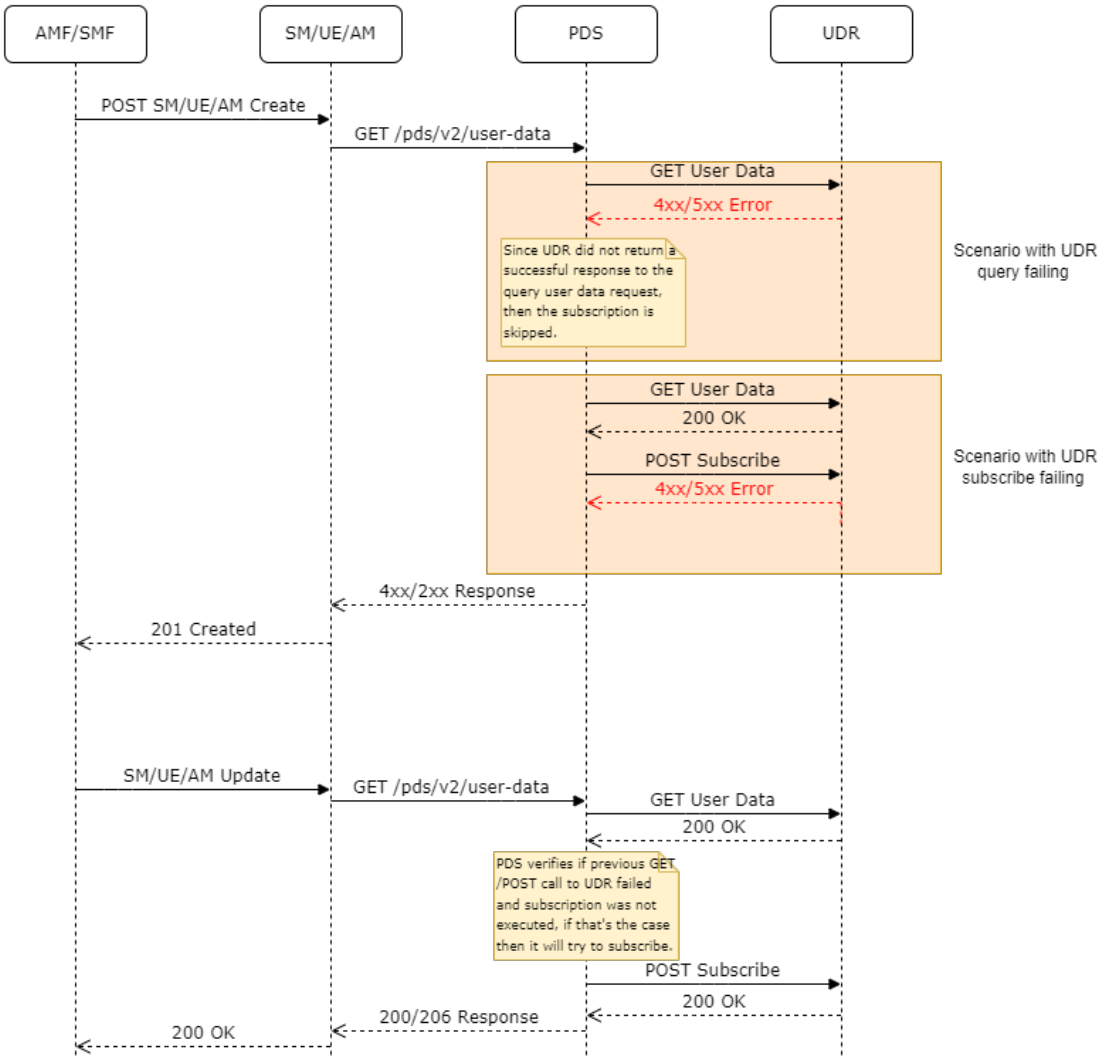
If `queryOnUpdate` and `subscriberToNotify` flags are enabled and an `update` request is received then PCF makes an Get Query request along and Post Subscriber request to the UDR.

That is,

- Policy should query and subscribe to UDR during Update request if `subscribeToNotify` and `queryOnUpdate` flags are enabled and the previous GET or POST requests failed (could be in the Create request or a previous Update request).
- Policy should not query UDR once GET or POST request is successful and subscription is successful.

Call Flow

Figure 4-54 Retrieving user data from UDR upon a GET or Update request to a Policy Service



1. Policy Service (SM Service/ AM Service/ UE Policy Service) receives a create request from SMF/AMF.
2. Policy Service sends a request to PDS to get the user data.
3. PDS sends a GET request to UDR to retrieve the data.
4. If the user data does not exist, UDR responds with an error and subscription to user data from UDR is also not performed.

5. If the user data is found, UDR responds with a 200 OK message.
6. PDS sends a POST Subscription to UDR to subscribe for any update to the user data.
7. The respective Policy service (SM Service/ AM Service/ UE Policy Service) creates an association for the request and responds to SMF/AMF.
8. When the Policy service (SM Service/ AM Service/ UE Policy Service) receives an update request from SMF/AMF, the service do not send a GET request to PDS to retrieve the user data from UDR with `queryOnUpdate` and `subscribeToNotify` flags are enabled.
9. PDS sends the GET request to UDR and receives a 200 OK response.
10. If the previous GET or POST Subscription request to UDR has failed, PDS sends a POST subscription request to UDR once again to subscribe for notification on any update to user data in UDR.
11. PDS forwards the successful response to the respective Policy Service (SM Service/ AM Service/ UE Policy Service), which in turn forwards the response to SMF/AMF.

PDS Handling Subscription Failure from UDR

If UDR subscription fails with subscribeToNotify flag enabled, the user must configure the following advanced settings keys under PDS Settings page on CNC Console to reattempt the subscription call on next SM update.

- NOTIFY_PCF_SM_ON_SUBSCRIPTION_FAILURE
- SUBSCRIPTION_FAILURE_NOTIFICATION_DELAY
- SUBSCRIPTION_FAILURE_NOTIFICATION_RETRY_COUNT
- SUBSCRIPTION_FAILURE_NOTIFICATION_RETRY_DELAY

For more information, see PDS Settings.

> **Note**
>
> In this case, queryOnUpdate flag for SM Service must be set to false.

With these configurations, PCF sends the subscription request towards UDR on receiving subscription failure notification from SM service.

> **Note**
>
> For AM and UE Policy services, UDR subscription request is reattempted using queryOnUpdate flag.

If PDS receives 404 status code from UDR, it performs user-level clean up for core services and session-level clean up for Pcrf-Core.

When revalidation and resetContext flags are set to false and the request is same for the existing context without the subscription information and the fetch request from the data source is successful, PDS updates the database with subscriber and context information.

The core services (SM, UE, or AM services) have flags to persist subscription failure. The flag is set to false when GET operation is performed and set to true when subscription failure notification is received from PDS.

A new property called "subscriptionFailure" is added to the JSON schema of SmPolicyAssociation. This property holds an integer value, whether there is a subscription failure or not. This property value is determined by setting some specific bits in the integer number. The following table shows which bit specifies the type of data subscription failure:

Table 4-22 SmPolicyAssociation Data Type

Data Type	Bit Number
smData	1
operatorSpecificData	2

Managing the Query on Update feature

Enable

You can enable the Query on Update feature using CNC Console or REST API for Policy.

- **Enable using CNC Console**:
  - To enable the feature for SM Service, set the value of **Query User on Update** parameter to `true` under **User** group on **PCF Session Management** page.

    For more information, see PCF Session Management.
  - To enable the feature for AM Service, set the value of **Query User on Update** parameter to `true` under **User** group on **PCF Access and Mobility** page.

    For more information, see PCF Access and Mobility.
  - To enable the feature for UE Policy Service, set the value of **Query User on Update** parameter to `true` under **User** group on **PCF UE Policy Service** page.

    For more information, see PCF UE Policy Service.
- **Enable using REST API**:
  - Set the value of **queryUserOnUpdate** parameter under **{apiRoot}/oc-cnpolicy-configuration/v1/services/pcfsm** API to `true`.

    For more information, see *Session Management Service* section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.
  - Set the value of **queryUserOnUpdate** parameter under **{apiRoot}/oc-cnpolicy-configuration/v1/services/pcfam** API to `true`.

    For more information, see *Access and Mobility Service* section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.
  - Set the value of **queryUserOnUpdate** parameter under **{apiRoot}/oc-cnpolicy-configuration/v1/services/pcfue** API to `true`.

    For more information, see *UE Policy* section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

You can configure the Query on Update feature using the CNC Console or REST API for Policy.

- **Configure using CNC Console**: To configure Query on Update feature for SM Service, AM Service and UE Policy service, perform the feature configurations under the User group on the respective service configurations page. For more information, see the following sections:

  - PCF Session Management
  - PCF Access and Mobility
  - PCF UE Policy Service
- **Configure using REST API**: Policy provides the following REST API to configure the Query on Update feature:
  - **{apiRoot}/oc-cnpolicy-configuration/v1/services/pcfsm**
  - **{apiRoot}/oc-cnpolicy-configuration/v1/services/pcfam**
  - **{apiRoot}/oc-cnpolicy-configuration/v1/services/pcfue**

Observe

The following PDS metrics provide the information about query on update feature:

- server_request_total
- server_response_total

For more information, see Policy DS Metrics.

4.37 Support for Presence Reporting Area

PCF supports the Presence Reporting Area (PRA) functionality to comply with the 3GPP standards.

Policy supports PRA functionality for SM Service, AM Service and UE Policy Service.

The Presence Reporting Area is an area defined within the 3GPP packet domain for reporting the presence of UE within that area. This is required for policy control and in charging scenarios. In E-UTRAN, the PRA can consist in a set of neighbor or non-neighbor Tracking Areas, or eNBs or cells.

There are two types of Presence Reporting Areas:

- UE-dedicated Presence Reporting Areas: Defined in the subscriber profile and composed of a short list of TAs and/or NG-RAN nodes and/or cells identifiers in a PLMN.
- Core Network pre-configured Presence Reporting Areas: Predefined in the AMF and composed of a short list of TAs and/or NG-RAN nodes and/or cells identifiers in a PLMN.

If the PRA feature is supported and SMF or AMF receives the PRA information from the serving node indicating any of the following:

- the UE is inside or outside of one or more PRAs
- any of the presence reporting areas is set to inactive

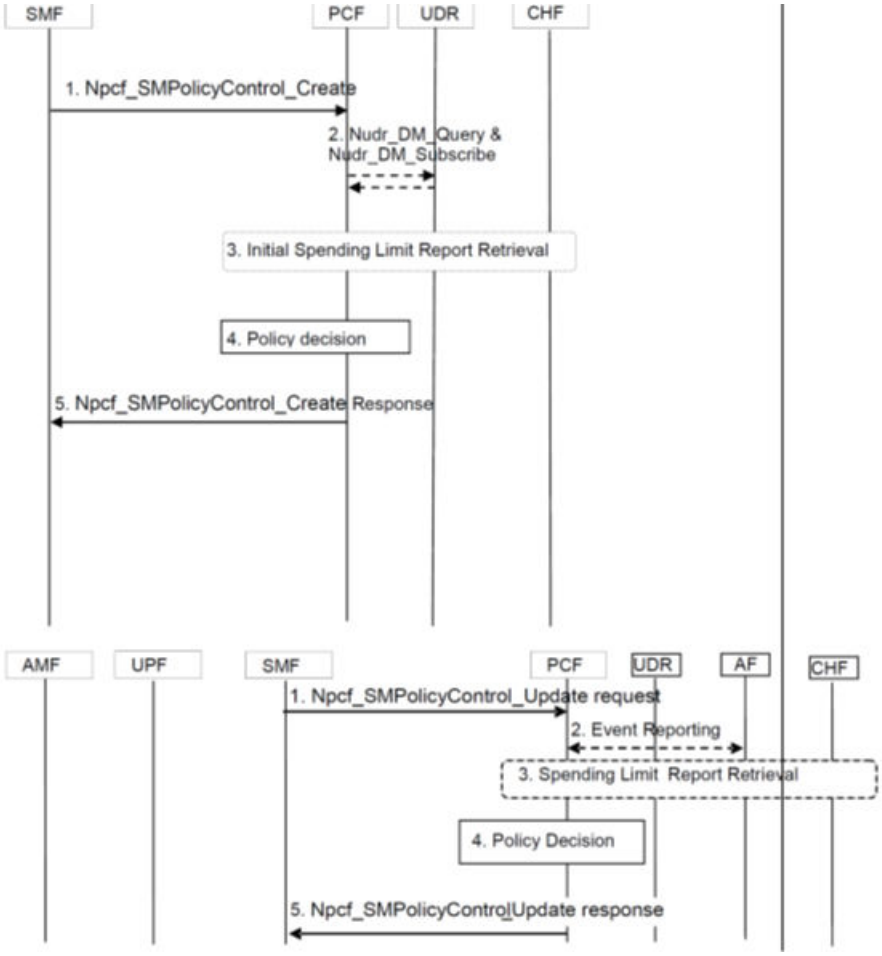
the SMF or AMF checks if the reported PRA identifier corresponds to a PRA that is relevant for the PCF. In that case, the SMF or AMF within the `SmPolicyUpdateContextData` or `AmPolicyUpdateContextData` or `UEPolicyUpdateContextData` data structure includes the `PRA_CH` within the `repPolicyCtrlReqTriggers` attribute and one or more PRA information report within the `repPraInfos` attribute. For each `PresenceInfo` data structure, the SMF or AMF includes the PRA status within the `presenceState` attribute and the PRA identifier within the `praId` attribute for each of the presence reporting areas reported by the serving node.

If the SMF or AMF receives additional presence reporting area information along with the PRA Identifier, the SMF or AMF only provides the PCF with the presence reporting area information corresponding to the additional PRA information.

- The SMF or AMF receives additional presence reporting area information when the UE enters or leaves one or more presence reporting areas related to a PRA set. In that case, the additional presence reporting area information corresponds to the actual individual presence reporting area. The received presence reporting area identifier corresponds to the PRA set ID and is used to identify the requester (PCF or CHF) of the notification information.
- PCF can acquire the necessary data for presence reporting from the UDR.
- Homogeneous support of PRA in a network is assumed.
- The serving node can activate the reporting for the PRAs which are inactive as described in the 3GPP TS 23.501 [2].

The following figure shows a call flow for PRA functionality for SM Service:

Figure 4-55 PRA Support for SM Service



The call flow is described as follows:

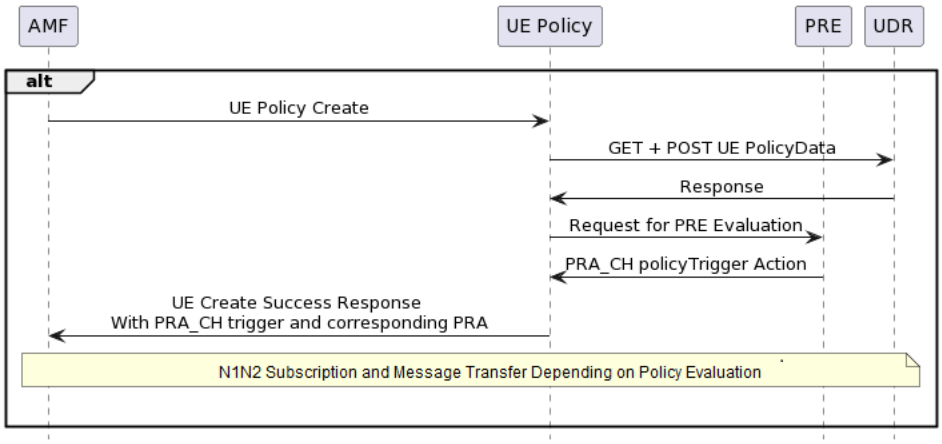
1. UE initiates a PDU session establishment request.
2. SMF sends an *Npcf_SMPolicyControl_Create* request message to PCF to establish an SM policy control correlation including the PDU session related information.
3. PCF completes the policy calculation based on the policy configuration and subscription information and user information. The PCF sends an *Npcf_SMPolicyControl_Create*

response message to the SMF, including an SM policy and PRA information. The PRA type can be either UE-dedicated PRA or Set of Core Network PRA.

> **Note**
>
> Currently, UE-dedicated PRA is not supported for SM Service.

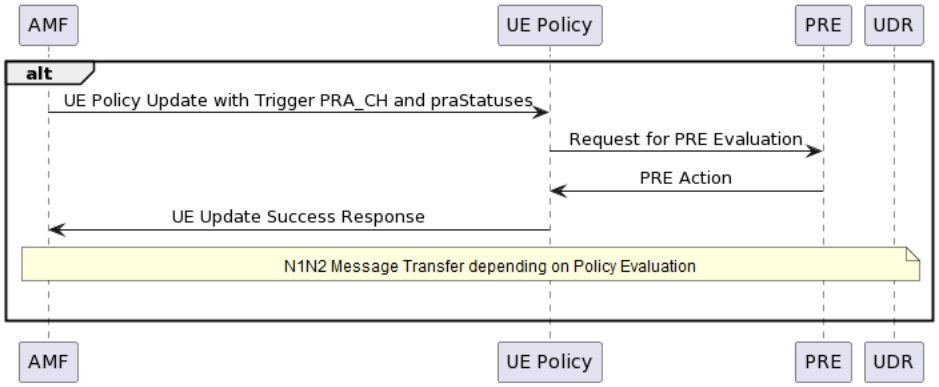
4. When the user changes across the PRA, the SMF sends an *Npcf_SMPolicyControl_Update* request message to PCF to modify the SM policy control correlation, and report the change in location of the user to the PCF. The message contains the information related to change of UE presence in Presence Reporting Area.
5. The PCF sends an *Npcf_SMPolicyControl_Update* response message to the SMF, including the updated SM policy.

Figure 4-56 UE Policy Association create scenario:



1. UE Policy Service receives a UE Policy Association create request from AMF.
2. UE Policy Service sends a GET request to PDS along with a POST subscribe to fetch the user data from UDR.
3. PDS sends the request to UDR via the UDR Connector.
4. UDR responds with the user data as well as the triggers for UE location change and PRA change.
5. PDS forwards the details to UE Policy Service.
6. UE Policy Service sends the details to PRE for evaluation.
7. PRE evaluates and installs the PRA successfully.
8. If there is a Policy Evaluation that requires N1/N2 Policy transfer, AMF subscribes and transfers N1N2 messages.

Figure 4-57 UE Policy Association update scenario



1. UE Policy Service receives a UE update request from AMF which includes the UE Policy update along with PRA_CH and praStatuses.
2. UE Policy service sends the updates to PRE for evaluation.
3. PRE evaluates the details and triggers the appropriate action on the PRA. '
4. UE Policy Service responds to AMF with a UE Policy update successful message.
5. If there is a Policy Evaluation that requires N1/N2 Policy transfer, AMF transfers N1N2 messages.

Managing Presence Reporting Area

Enable

The PRA functionality is a part of Policy configurations. You do not need to enable or disable this feature.

Configure

You can add new PRAs and configure the them using the CNC Console or REST API for Policy.

- **Configure using CNC Console:** Perform the configurations on the **PCF Presence Reporting Area** page. For more information about configuring PRAs, see PCF Presence Reporting Area.
- **Configure using REST API:** Policy provides the following REST API for PRA configuration:
  API: {apiRoot}/ocpm/pcrf/v1/configuration/policy/pra

  You can perform the POST, PUT, or GET operations to configure the PRA feature. For more information about REST API configuration, see "PRA" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

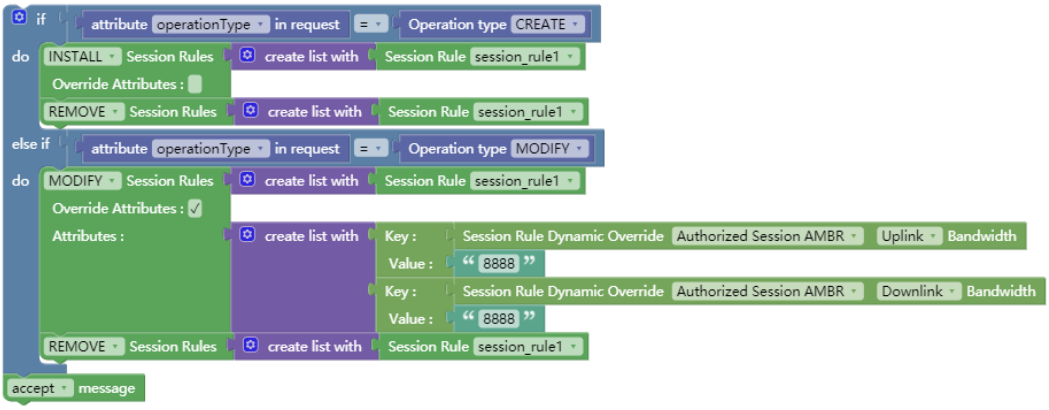
4.38 Handling Install and Remove Conflict for Same Rule

This section describes a high-level design of the Conflict Resolution of INSTALL and REMOVE actions on the same PCC/Session Rules when the remove action is **Remove** (ALL, Predefined, Dynamic, Conditioned, non-conditioned) in the policy project.

In certain policies, install/update and remove actions are applied for the same rule - which can be either a session rule or a PCC rule.

The following screen capture illustrates a policy to show conflicts in same session rule. In this example, the install, update, and remove actions are applied to the same session rule, that is, session_rule1:

Figure 4-58 Example of Policy with Conflicts in a Session Rule



In previous releases, after executing a similar policy, SM service would remove the rule. However, with this feature, PCF is enhanced to configure the precedence and resolve the conflicting install and remove actions on the same rule.

When SM service receives two action types, such as install and remove for the same rule, SM service processes the remove action for the first rule, and then installs it. However, if install is not configured for the rule, SM service can remove it.

Policy allows its users to select how to resolve such conflicts in the most suitable manner by using the `Install/Remove Rule Conflicts Strategy` parameter under PCF Session Management service configurations.

Managing Install/Remove Rule Conflicts Handling

Enable
The Handling Install/Remove Rule Conflicts is the core functionality of Policy configurations. You do not need to enable or disable this feature.

Configure
You can configure the managing Rule conflict functionality using the CNC Console or REST API for Policy.

- **Configure using CNC Console:** Set value of the **Install/Remove Rule Conflicts Strategy** parameter under the **Rule** group on the PCF Session Management page. For more information, see configurations for **Rule** in PCF Session Management.
- **Configure using REST API:** Policy provides the following REST API for Install/Remove Rule Conflicts configuration:
  API: {apiRoot}/oc-cnpolicy-configuration/v1/services/pcfsm

  Set value of the **ruleConflictsStrategy** parameter in the **Session Management Service** API. For more information about REST API configuration, see Rule Configurations of the

"Session Management Service" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

4.39 Diameter Session Retry

In PCF, diameter connector sends the request messages such as Rx RAR (Re-Authorization Request) or Rx ASR (Abort Session Request) via the diameter gateway to Application Function (AF). AF acknowledges these commands by sending a successful or failed messages Re-Authorization Answer (RAA) or Abort Session Answer (ASA) respectively to PCF.

The RAA or ASA messages answers when received with protocol errors such as 3002 (DIAMETER_UNABLE_TO_DELIVER) or 3004 (DIAMETER_TOO_BUSY) or session timeout, the operator would want to retry sending the same Rx RAR or Rx ASR diameter messages from PCF to AF through a different or an alternate BSF (Binding Support Function) or DRA (Diameter Routing Agent).

The Diameter Message Retry for Rx RAR and Rx ASR diameter messages is enabled through the Error Mapping Framework feature in the diameter gateway service. This framework resolves application errors and takes necessary action based on the error context. The error handler framework tries to find an alternate solutions based on the configurations set in CNC Console. If the error is resolved, it sends back the success result to the caller, else it either retries based on the configured maximum resolution attempts number or terminates by forwarding the last known error.

The operator should have configured a host and realm for retry of failed diameter messages in the diameter routing table. Diameter Gateway finds the alternate peer from the routing table. If the diameter routing table is not configured then there is no retry behavior from policy.

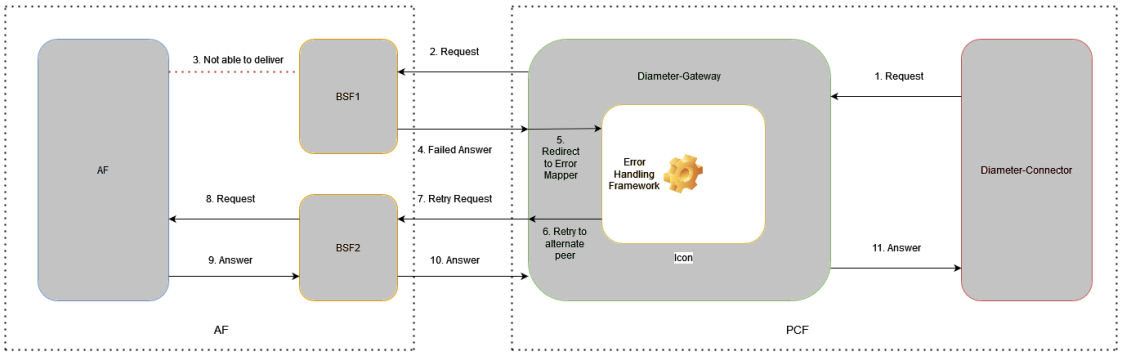
> **Note**
>
> Default number of retry attempts is one, when the alternate peers are available. If alternate peers are not available, no retry attempt is made.

PCF retries resending Rx-RAR and Rx-ASR diameter messages for the following configurable error code series:

- 3xxx (Protocol Errors)
- 4xxx (Transient Failures)
- 5xxx (Permanent Failure)

For more information about these error codes refer to Diameter Error Codes feature section.

Figure 4-59 The block diagram represents the diamater Rx RAR and Rx ASR message flow with a retry:



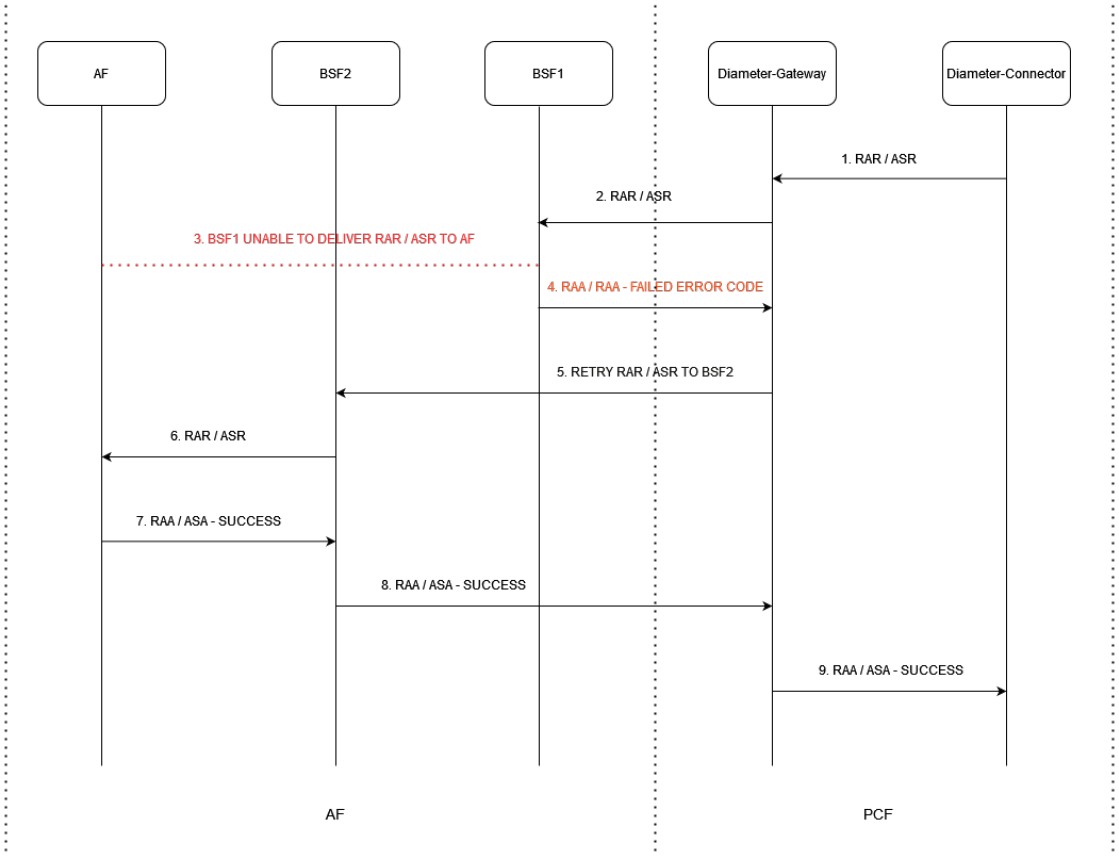
The block diagram represents the diamater Rx RAR and Rx ASR message flow with a retry using the Error Handler Framework in Policy.

1. The diameter connector sends the diameter request and the diameter gateway receives the diameter request.
2. The diameter gateway forwards this diameter request to available BSF1 based on routing table and priority.
3. The BSF1 is not able to deliver this message to AF due to some reason.
4. The diameter gateway receives the failed diameter answer from the BSF1 then it redirects the request context to Error Handler.
5. Error Handler gets the request context from the failed diameter request and finds the alternate retry action configured.
6. The Error Handler provides the retry action for the same request to next available alternate BSF2.
7. The diameter gateway sends the retry diameter request to next alternate BSF2.
8. BSF2 sends the retry request to AF.
9. AF sends the answer to BSF2 back.
10. BSF2 sends the answer to diameter gateway.
11. The diameter gateway forwards the answer received from BSF2 to the diameter connector.

Session Retry Call Flows

The following call flow describes a successful transmission of Rx RAR or Rx ASR message to AF.

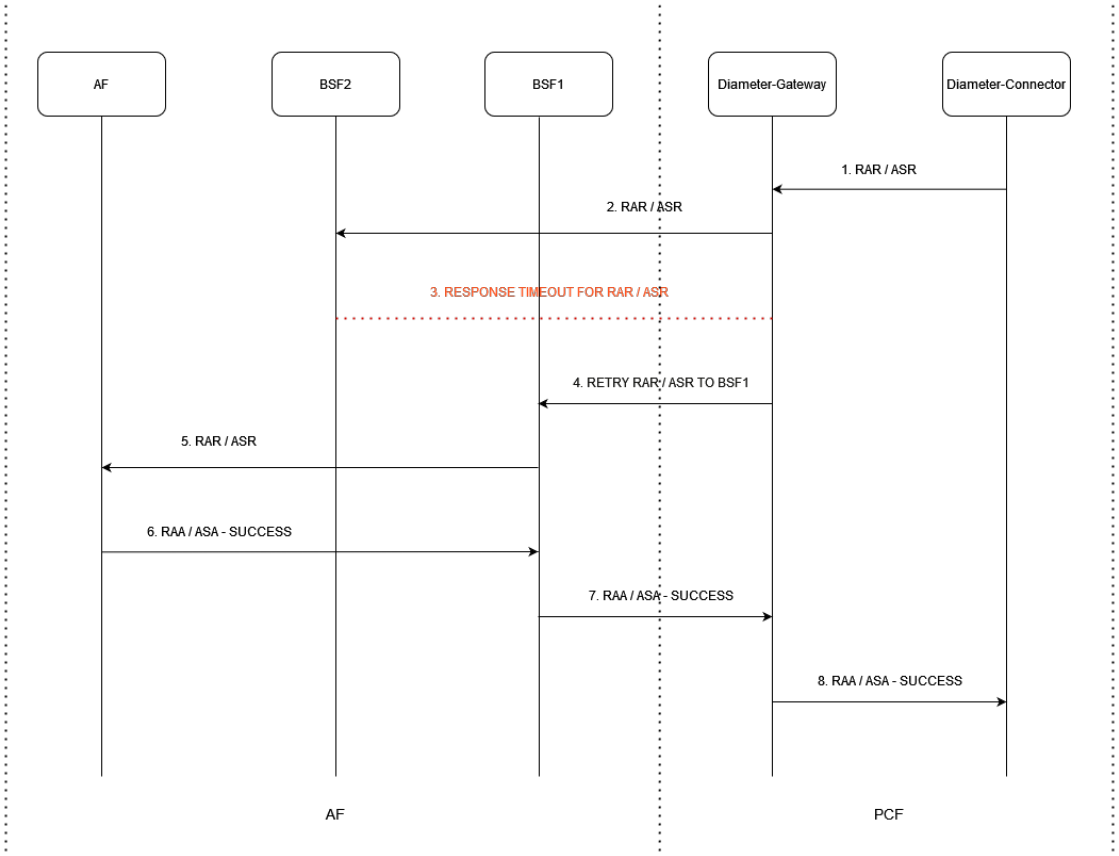
Figure 4-60 Call Flow Rx RAR or Rx ASR one time message retry - Success Case:



1. Diameter connector sends RAR or ASR request to diameter gateway.
2. Diameter Gateway sends the request to the available BSF, here to BSF1.
3. BSF1 is unable to deliver the RAR or ASR message to AF.
4. BSF1 returns a failed RAA or ASA message along with an error code to diameter gateway.
5. Diameter Gateway interprets the error code, and retries to send the failed message to the next available BSF, here BSF2 gets the message.
6. BSF2 sends the RAR or ASR message successfully to AF.
7. AF returns successful RAA or ASA message response to BSF2.
8. BSF2 sends this successful response to diameter gateway.
9. Diameter Gateway sends this response to diameter connector.

The following call flow describes a successful retry in case of a response timeout for Rx RAR or Rx ASR.

Figure 4-61 Call Flow RX RAR or RX ASR Timeout Retry - Success Case:



1. Diameter connector sends RAR or ASR request to diameter gateway.
2. Diameter Gateway sends the request to the available BSF, here to BSF1.
3. BSF1 is unable to deliver the RAR or ASR message to AF.
4. BSF1 returns a failed RAA or ASA message along with an error code to diameter gateway.
5. Diameter Gateway interprets the error code, and retries to send the failed message to the next available BSF, here BSF2 gets the message.
6. BSF2 sends the RAR or ASR message successfully to AF.
7. AF returns RAA or ASA message successful response to diameter gateway.
8. Diameter Gateway sends this response to diameter connector.

Retry Attempts

The user can choose to configure the number of retries to find a Peer for sending the Rx-RAR and Rx-ASR diameter messages. This retries is possible only when the alternate peers are available i.e., the Peer is alive, if it is not available i.e., the Peer is not alive then there is no retry attempt made. The default number of retry attempts is 1. The number of retry value can be any positive number between 1 to 2147483647.

The number of retries can be set through the advance settings configurations using the following advance setting keys:

- DIAMETER.ErrorHandler.MaxRetryCount.Rx.RAR

- DIAMETER.ErrorHandler.MaxRetryCount.Rx.ASR

If the above advance settings configurations are not supported in the CNC Console, then the default retry attempt is 1. If retry attempt value configured is a zero or negative number, then the default retry attempt value shall be considered.

Peer Cycle Back Retry

The user can choose to perform Peer retry in a cyclic manner if the number of retries configured is more than the number of alternate peers available for both Rx RAR and RX ASR message retries. This retries is possible only when the alternate peers are available i.e., the Peer is alive, if it is not available i.e., the Peer is not alive, then there is no retry attempt made. The user configures this in CNC Console by setting `retry peer cycle back` field to true.

This peer cycle back retry configuration is performed using the following advance setting keys:

- DIAMETER.ErrorHandler.CycleBackRetry.Rx.RAR
- DIAMETER.ErrorHandler.CycleBackRetry.Rx.ASR

If above advance settings configurations are not supported in the CNC Console, then the default peer cycle back retry is false.

Described below are the scenario that shows how the Diameter message retry mechanism works.

For example 1: Consider, the number of configured retries is 2 and only 2 BSF/DRA are configured as alternate peers in PCF. Both the peers are alive.

- The Rx RAR message is sent via (PCF-1) - (BSF1/DRA1) - (PCSCF-1) but the response has failed with error code 5012/timeout/3002/3004
- Then the message is resent (retry 1) via (PCF-1) - (BSF2/DRA2) - (PCSCF-1) but the response has failed with error code 5012/timeout/3002/3004.
- Then the message is resent (retry 2) again via (PCF-1) - (BSF1/DRA1) - (PCSCF-1) but the response has failed with error code timeout.

During the second time, the message is resent using the first peer again as only 2 BSF/DRA are available and alive, and peer cycle back retry mechanism is applied.

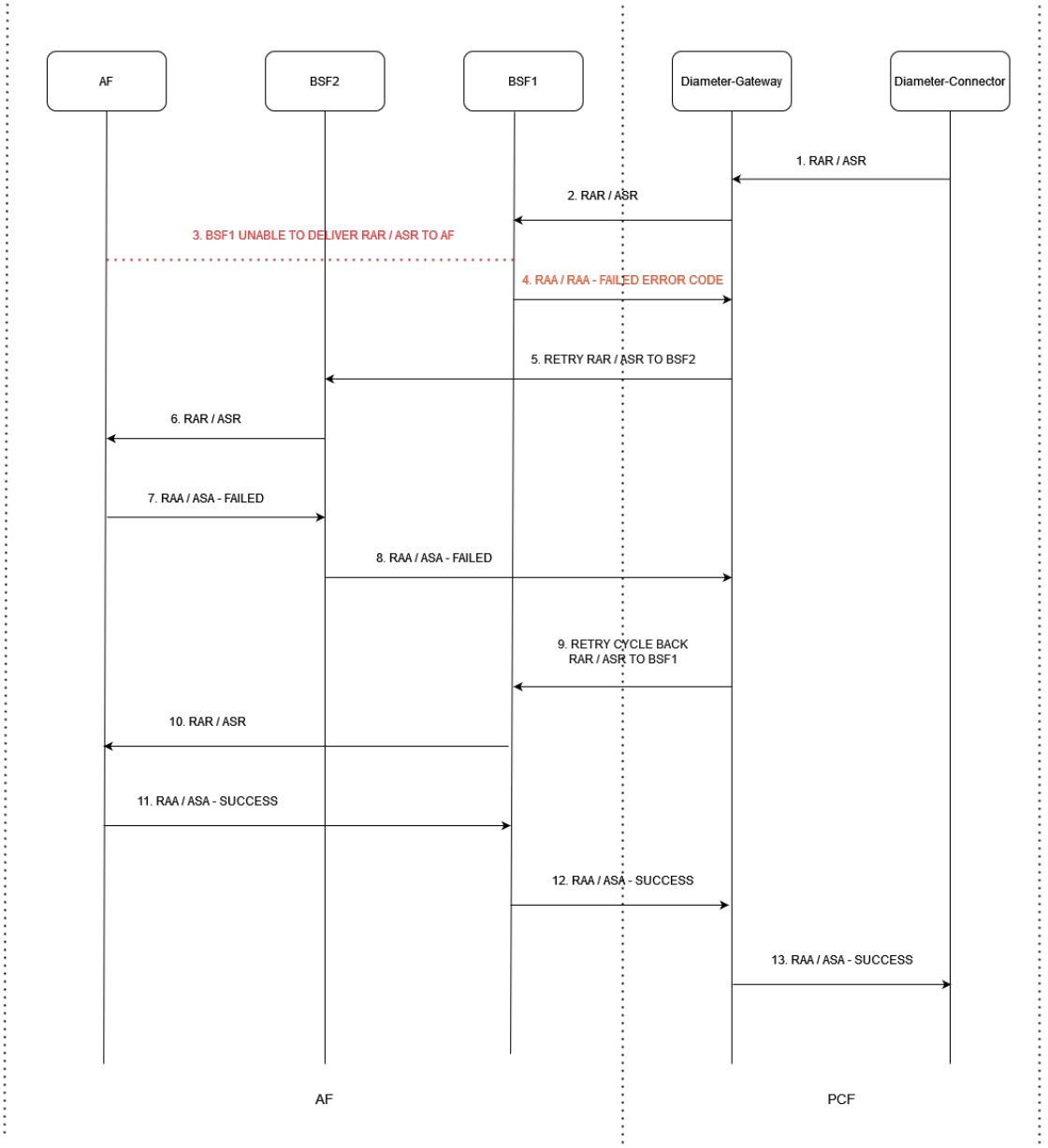
For example 2: Consider that the number of configured retries is 3 and only 2 BSF/DRA are configured as alternate peers in PCF and just one of the Peer BSF/DRA is alive.

- The Rx RAR message is sent via (PCF-1) - (BSF1/DRA1) - (PCSCF-1) but the response has failed with error code 5012/timeout/3002/3004
- Then the message is resent (retry 1) via (PCF-1) - (BSF1/DRA1) - (PCSCF-1) but the response has failed with error code 5012/timeout/3002/3004.
- Then the message is resent (retry 2) again via (PCF-1) - (BSF1/DRA1) - (PCSCF-1) but the response has failed with error code timeout.

The message is resent 3 times through the same peer as only one BSF/DRA is alive, and cycle back retry mechanism is applied

The following call flow describes the peer cycle back retry when the number of retry count is higher than the alternate peers available.

Figure 4-62 RX RAR/RX ASR Retry - Peer Cycle Back When Retry Count > Alternate Peers



1. Diameter connector sends ASR or RAR request to the diameter gateway
2. Diameter Gateway sends the request to the available BSF, here to BSF1.
3. BSF1 is unable to deliver the RAR or ASR message to AF.
4. BSF1 returns a failed RAA or ASA message along with an error code to diameter gateway.
5. Diameter Gateway tries to resend the failed message to the next available BSF, here BSF2 gets the message.
6. BSF2 sends the RAR or ASR message successfully to AF.
7. AF responds with failed RAA/ASA response to BSF2.
8. BSF2 sends the failed RAA/ASA response to diameter gateway.

9. At Diameter Gateway, if retry cycle back is enabled, then it resend the RAR or ASR message back to BSF1.
10. BSF1 sends the RAR or ASR message successfully to AF.
11. AF returns RAA or ASA message successful response to BSF1.
12. BSF1 returns RAA or ASA message successful response to diameter gateway.
13. Diameter Gateway returns RAA or ASA message successful response to diameter connector.

Error Originator Peer

The Error Originator Peer indicates the peer host where the Rx RAR or Rx ASR failed message error occurred/originated while sending or retry sending the Diameter messages. The user can customize the error origination peer by using `Error Originator` filed in the CNC Console and the customizing options are based on:

- An error received from an intermediate peer (INTERMEDIATE PEER).
- An error received from the destination peer, which is not an intermediate peer (DESTINATION PEER).
- An error received from any peer (ANY).

An intermediate peer is BSF/DRA and a destination peer (not intermediate peer) is P-CSCF.

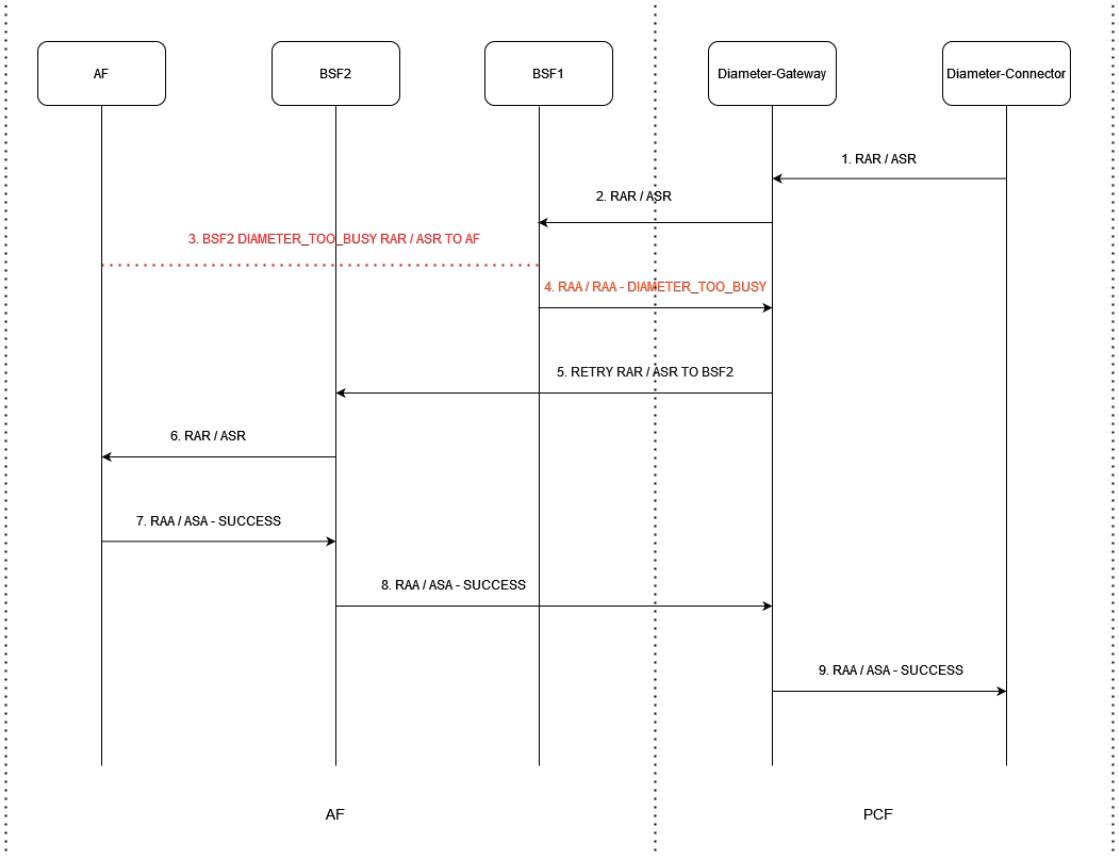
By default, the error originator peer option is any peer. If multiple error originator peers are configured for the same error code, then the first priority error originator configurations, applies for the response failures.

In case of response timeout, the error originator would be considered based on:

- The destination host in RAR/ASR is af.xxx.com. If message is sent to bsf.xxx.com and if response timeout happens, then it is considered as error originated from intermediate peer.
- The destination host in RAR/ASR is af.xxx.com. If message is sent to af.xxx.com and if response timeout happens, then it is considered as error originated from destination peer.

The following call flow describes the error origination by intermediate peer.

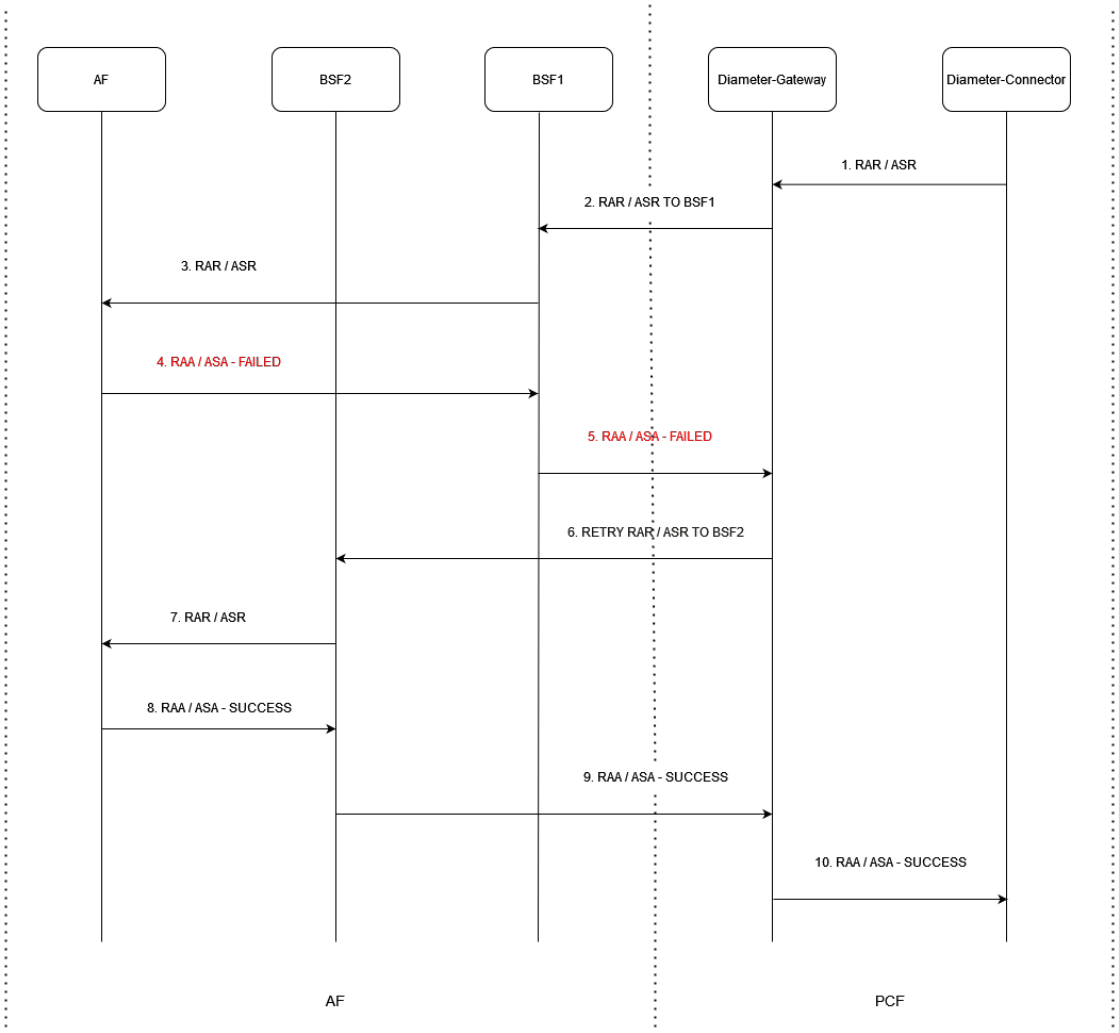
Figure 4-63 RX RAR/RX ASR Retry - Error Originated by Intermediate Peer



1. Diameter connector sends ASR or RAR request to the diameter gateway.
2. Diameter Gateway sends the request to the available BSF, here to BSF1.
3. BSF1 gets DIAMETER TOO BUSY error.
4. BSF1 forwards this error to diameter gateway.
5. Diameter Gateway retries sending this failed message through BSF2.
6. BSF2 sends this message to AF.
7. AF returns RAA or ASA message successful response to BSF2.
8. BSF2 returns RAA or ASA message successful response to diameter gateway.
9. Diameter Gateway returns RAA or ASA message successful response to diameter connector.

The following call flow describes the error origination by destination peer (not intermediate peer).

Figure 4-64 RX RAR/RX ASR Retry - Error Originated by Destination Peer (Not Intermediate Peer)



1. Diameter connector sends ASR or RAR request to the diameter gateway.
2. Diameter Gateway sends the request to the available BSF, here to BSF1.
3. BSF1 sends the RAR or ASR message successfully to AF.
4. AF (Destination Peer) sends the failed RAA/ASA response to BSF1.
5. BSF1 sends the failed RAA/ASA response to diameter gateway.
6. Diameter Gateway retries sending this failed message through BSF2.
7. BSF2 sends this message to AF.
8. AF returns RAA or ASA message successful response to BSF2.
9. BSF2 returns RAA or ASA message successful response to diameter gateway.
10. Diameter Gateway returns RAA or ASA message successful response to diameter connector.

Default Error Handling Configuration

PCF provides the default error handling configuration to attempt diameter session retry on all error codes (except diameter result code 2xxx) and timeout for Rx RAA and Rx ASA failed diameter messages. When the diameter message retry feature is enabled on Rx interface, these default error handling configurations get applied by default. The user has an option to enable/disable these default configurations through the CNC Console edit configurations.

The error configuration consists of error state and error cause. Error state, a mandatory parameter provides the details about the error that occurred. Error cause, a optional parameter provides additional information on the error occurred such as sub-status, error message. The error configuration can be assigned a priority of evaluation in case the error matches with more than one error state.

For all default error handling configurations, the value for retry attempt is 1 and peer cycle back retry is false. The number of retry value can be any positive number between 1 to 2147483647.

> **Note**
>
> If retry attempt value configured is a zero or any negative number, then the default retry attempt value shall be considered.

A sample default configuration looks like:

- Message - Rx-RAR, Status code - ANY (except 2001)
- Message - Rx-ASR, Status code - ANY (except 2001)
- Message - Rx-RAR, Status code - ANY, Error Cause Message - TIMEOUT_EXCEPTION
- Message - Rx-ASR, Status code - ANY, Error Cause Message - TIMEOUT_EXCEPTION

Managing Diameter Session Retry

This section explains the procedure to enable and configure the feature.

Enable

By default, Diameter Message Retry behavior is disabled for Rx interface and operator can enable this feature through the CNC Console configurations.

Configure Using CNC Console

Perform the feature configurations in CNC Console as described in Error Configurations section.

Configure Using REST API

Perform the export/import error configurations as described in "Error Configurations" section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Document*

Observability

Metrics

Following metrics were updated in the Diameter Gateway Metrics section:

- `occnp_diam_request_local_total`
- `occnp_diam_request_network_total`

- `occnp_diam_request_inter_total`

Alerts

Following alerts are used by this feature:

- RAA_RX_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD
- RAA_RX_FAIL_COUNT_EXCEEDS_MAJOR_THRESHOLD
- RAA_RX_FAIL_COUNT_EXCEEDS_MINOR_THRESHOLD
- ASA_RX_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD
- ASA_RX_FAIL_COUNT_EXCEEDS_MAJOR_THRESHOLD
- ASA_RX_FAIL_COUNT_EXCEEDS_MINOR_THRESHOLD

Maintain
If you encounter alerts at system or application levels, see Alerts section for resolution steps.

In case the alerts still persist, perform the following:

- Collect the logs: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide*.
- Raise a service request: See My Oracle Support for more information on how to raise a service request.

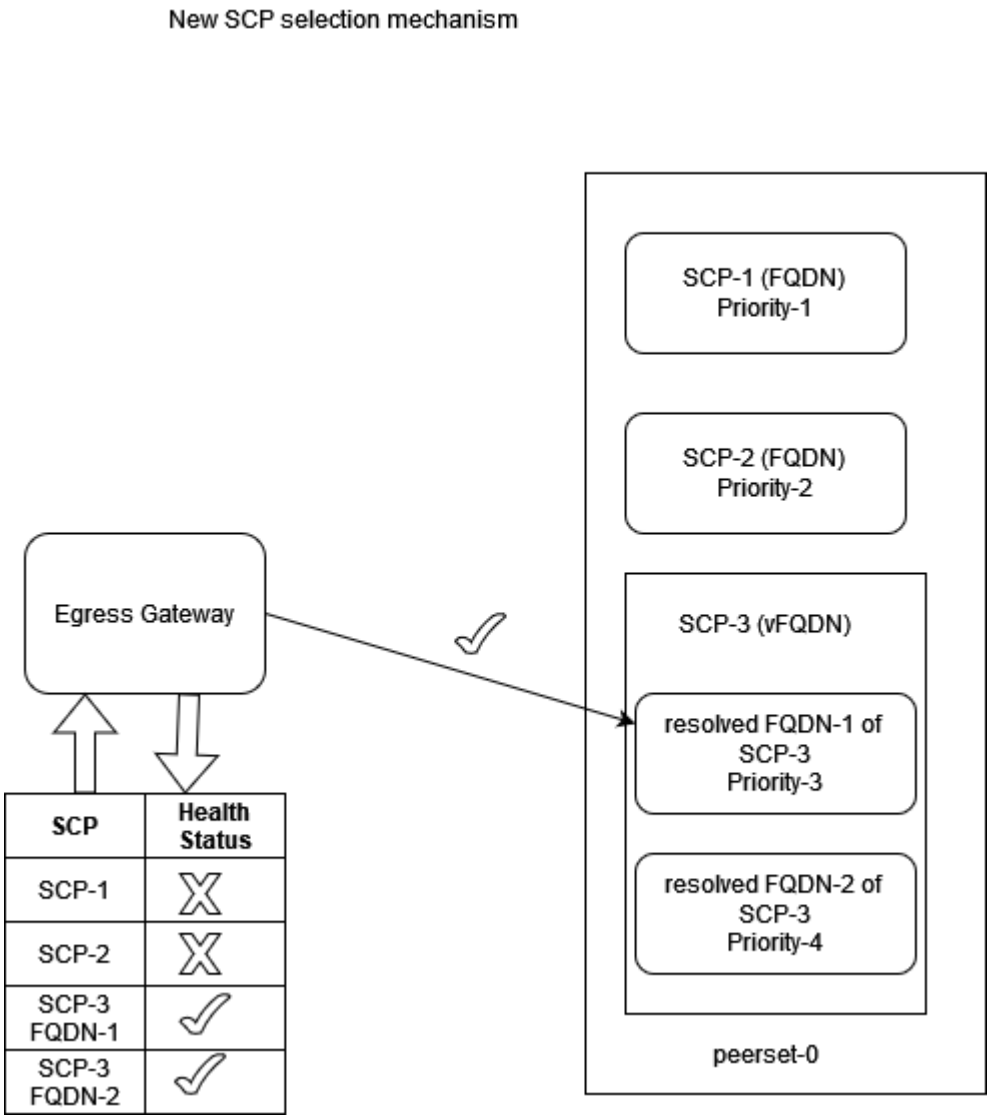
4.40 Monitoring the Availability of SCP using HTTP2 OPTIONS

Policy determines the availability and reachability status of all SCPs irrespective of the configuration types.

This feature is an enhancement to the existing SBI routing functionality. Egress Gateway microservice interacts with SCP on their health API endpoints using HTTP2 OPTIONS method. It monitors the health of configured SCP peers to ensure that the traffic is routed directly to the healthy peers. This enhancement avoids routing or rerouting towards unhealthy peers, thus minimizing the latency time.

Egress Gateway microservice maintains the health status of all available and unavailable SCPs. It maintains the latest health of SCPs by periodically monitoring and uses this data to route egress traffic to the most preferred healthy SCP.

Figure 4-65 New SCP selection mechanism



Once `peerconfiguration, peersetconfiguration, routesconfiguration, and peermonitoringconfiguration` parameters are configured at Egress Gateway microservice, and all SCPs (after Alternate Route Service (ARS) resolution, if any vFQDN is configured) are marked initially as healthy. The peers attached to the associated peerset are scheduled to run health API checks and update the health status continuously.

During the installation, the value of the parameter `peermonitoringconfiguration` is set to false by default. Since, this feature is an add-on to the existing SBI Routing feature and will be activated if the `sbirouteconfig` feature is enabled. To enable this feature, perform the following:

- configure `peerconfiguration` with `healthApiPath`
- configure `peersetconfiguration`
- configure `sbiroutingerroractionsets`
- configure `sbiroutingerrorcriteriasets`

- configure `routesconfiguration`
- enable `peermonitoring`

If SBI Routing feature is enabled before upgrading, the `healthApi` in `peerconfiguration` should be attached manually to existing configured peers. If the operator tries to enable peermonitoringconfiguration and the targeted peers do not have the `healthApiPath` then an appropriate error response is sent.

Managing Monitoring the Availability of SCP Using SCP Health APIs

This section explains the procedure to enable and configure the feature.

Configure

You can configure the Monitoring the Availability of SCP using the REST API.

Configure Using REST API: Perform the following feature configurations as described in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Document*:

- create or update peer `Peer Configuration` with health status endpoint details.
- create or update the peerset `peersetconfiguration` to assign these peers
- enable the feature using the below peermonitoring configuration `peermonitoringconfiguration`.

> **Note**
>
> Health Monitoring of the peer will start only after the feature is enabled and the corresponding peerset is used in `sbirouteconfig`.

Observe

Following metrics are added in the CNC Policy Metrics section:

- oc_egressgateway_peer_health_status
- oc_egressgateway_peer_health_ping_request
- oc_egressgateway_peer_health_ping_response
- oc_egressgateway_peer_health_status_transitions
- oc_egressgateway_peer_count
- oc_egressgateway_peer_available_count

Alert

Following alerts are added in the Alert section:

- SCP_PEER_UNAVAILABLE
- SCP_PEER_SET_UNAVAILABLE

4.41 Supports 3gpp-Sbi-Correlation-Info Header

The 3gpp-Sbi-Correlation-Info header contains correlation information, User Equipment (UE) identity that is used by an operator in various offline network management, performance analysis, and troubleshooting tools and applications to identify messages (requests, responses, subscriptions, and notifications) related to a particular subscriber.

In CNC Console, enable this feature in the global configuration page. By enabling this, the correlation-info header gets applied across all of the PCF external interfaces like PCF Session Management(SM), PCF Access and Mobility(AM), PCF UE Policy, PCF User Connector and Gateways. The generation of new correlation-info headers and forwarding them to producer NF's is not managed by this global configuration UI option.

The generation of new correlation-info header in PCF is managed by the **NF Communication Profiles** configuration page. Communication profiles once attached with the respective interfaces (PCF SM, PCF AM, PCF UE Policy, and PCF User Connector) allows to enable or disable of header generation along with the flexibility of selecting correlation type to be used for the header. The Correlation types namely SUPI, GPSI or both are supported for this release. For more details about *NF Communication Profiles*, see the NF Communication Profiles section.

The following conditions are to be met for header generation in PCF:

1. Correlation-info header, not sent by consumer NF towards PCF interfaces.
2. CNC Console global settings configuration is enabled.
3. Communication profile attached to the respective interface has the required configuration.

Further, the received or generated headers are forwarded only when the setting **Send Correlation-Info Header** is set to Enable.

3gpp-Sbi-Correlation-Info

The header contains correlation information e.g., UE identifier related to the HTTP request or response.

> **Note**
>
> 1. The possibility to include more than 1 correlationinfo parameter in the 3gpp-Sbi-Correlation-Info header is kept for future extensibility.
>   correlationinfo = ctype "-" cvalue
>
>   ctype = "imsi" / "impi" / "suci" / "nai" / "gci" / "gli" / "impu" / "msisdn" / "extid" / "imei" / "imeisv" / "mac" / "eui" / token
> 2. The token is defined for future extensibility.
>   The token of ctype shall not use the dash ("-") character.
>
>   cvalue = 1*tchar

Table 4-23 The format of cvalue shall comply with the data type description.

ctype	Description
SUPI	VarUeId format defined for IMSI and starting after the string "imsi-"
GPSI	VarUeId format defined for MSISDN and starting after the string "msisdn-"

Table 4-24 3GPP defined Custom HTTP Headers

Header	Description	Example
3gpp-sbi-correlation-info	This header may be used to contain correlation information (e.g., UE identity), that may be used by an operator in various offline network management, performance analysis and troubleshooting tools/applications to identify messages (requests, responses, subscriptions, notifications) related to a particular subscriber.	EXAMPLE 1: When UE identifier used is SUPI and SUPI type is an IMSI: 3gpp-Sbi-Correlation-Info: imsi-345012123123123 EXAMPLE 2: When UE identifier used is GPSI and GPSI type is an MSISDN:3gpp-Sbi-Correlation-Info: msisdn-1234567890 EXAMPLE 3: When UE identifiers used are SUPI and GPSI where SUPI type is an IMSI and GPSI type is an MSISDN:3gpp-Sbi-Correlation-Info: imsi-345012123123123; msisdn-1234567890

Identifier Precedence for Generation of Correlation Header in request Toward other NFs

The UDR Connector service supports configurations to provide default user identifier that takes precedence over other identifiers. This ensures that these configured identifier should be considered while sending the requests toward external NFs like NRF or UDR. This key precedence `UDR.KeyPrecedence` value can be set by the user in the advanced settings of `PCF User Connector` service page in CNC Console. For more details, see the `Advanced Settings` in PCF User Connector section.

The configured precedence identifiers affects the generation of correlation header. The first identifier in the configuration shall be considered for the generation of the correlation header.

Consider a scenario where SBI Correlation header is configured and the default key precedence's are set to `IMSI,GPSI`. If both of the identifiers exists in the request then one of them takes priority based on the configured default key precedence. In this case IMSI shall be used for generating the correlation header while sending requests to external NFs UDR or NRF. If the precedence's are changed to `GPSI, SUPI` then for generation of correlation header GPSI is considered while sending the request to external NFs UDR or NRF.

Managing SBI messages correlation using Subscriber Identity

This section explains the procedure to enable and configure the feature.

Configure

In CNC Console, enable this feature by enabling `Enable SBI Correlation` field in the general settings page, see General Settings.

Configure Using REST API

For configuring the SBI messages correlation using Subscriber Identity feature using REST APIs, see *Oracle Communications Cloud Native Core, Policy REST API Specification Guide*.

Observe

Following metrics are added in Correlation-Info Header Metrics section:

- occnp_correlation_info_header_received
- occnp_correlation_info_header_forwarded
- occnp_correlation_info_header_generated

4.42 HTTP Error Codes

Policy can handle Protocol or Application errors and a few other additional defined errors for various scenarios. When Policy encounters an error in processing a request, it sends error codes in the response message to the request. With this enhanced functionality, Policy allows users to mitigate errors from different NFs.

Error Mapping for CHF and UDR Interfaces

Configure
In UDR and CHF, the session retry functionality is used to mitigate the errors. For more information, see Support for Session Retry and Alternate Route Service.

In addition to session retry, we can mitigate the errors using the object expressions with action as "Reject session with error code" blockly. If this blockly is not configured, PCF responds to SMF or AMF with "400" error code and "USER_UNKNOWN" cause code for any error response received from UDR while fetching subscriber profile. For more information, see "PCF-SM" section in Oracle Communications Converged Policy Design Guide.

Enable
To enable Error Handling feature where the Error Codes from CHF & UDR to reach the core services, the following flags needs to be updated in the Advanced Settings specific to the service:

- For UDR, **UDR_ errorHandlerEnabled** key must be set to true, in the advanced settings for PCF User Connector service.
- For CHF, **CHF_errorHandlerEnabled** key must be set to true, in the advanced settings for PCF User Connector service.
- For PDS, **PDS_ERROR_HANDLER_ENABLED** key must be set to true, in the advanced settings for PDS service.

For more information, see Configuring Policy Using CNC Console.

Observe
Added the following metrics for Error Mapping for CHF and UDR Interfaces :

- error_handler_in_total
- error_handler_out_total

> **Note**
>
> When the UDR Connector, CHF Connector, and PDS act as a consumer of the error and the Error Handler feature is integrated with these services and is enabled, and if these services are generating any error while interacting with other services, then the above metrics will increment. Whereas, When the UDR Connector, CHF Connector act as a producer of the error, the Error Handler feature is not considered. Hence, these metrics does not get incremented even when the feature is enabled.

For more information, see Error Mapping Metrics.

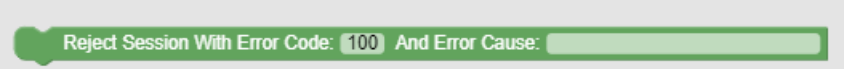
Error Mapping for AM, SM, and UE Interfaces

In AM, SM, and UE, the session retry functionality is used to mitigate some of the errors. For more information, see Support for Session Retry and Alternate Route Service.

Enable
This is a core functionality of Policy. You do not need to enable or disable this feature.

Configure
The following blockly has been added in the PCF-SM to reject a session with error code:

Figure 4-66 Reject Session with Error Code and Error Cause



This action blockly allow to reject the session with a custom error code and a custom error cause.

For more information, refer to Oracle Communications Cloud Native Core, Policy Design Guide.

Observe
The following metrics is incremented in case of error mapping for AM, SM, and UE interface:

- ocpm_ingress_response_total

For more information, see

Error Mapping for NRF Interface

In NRF, the Policy retries with primary and non-primary NRFs to configure the errors.

Enable
To enable the service, set the following parameter to true at the time of installing or upgrading Policy:

`notifySemanticValidationEnabled`

For information on the parameter, refer to Oracle Communications Cloud Native Core, Policy Installation, Upgrade, and Fault Recovery Guide .

Configure
This feature can be configured using the `notifySemanticValidationEnabled` parameter.

Observe
Added the following metrics for Error Mapping for NRF Interface :

- nrfclient_nw_conn_in_response_total
- nrfclient_nw_conn_out_notify_response_total

For more information, see NRF Client Metrics.

For the list and the details of the HTTP error codes of CHF, UDR, AM, SM, UE, and NRF interfaces, see HTTP Error Codes Supported by Policy.

4.43 Diameter Error Codes

Policy can handle protocol or application errors and a few other additional defined errors for various scenarios. When Policy encounters an error in processing a request, it sends error codes in the response message to the request. Policy interacts with BSF and at Rx interface during the life of a session there could be an error scenario.

Policy should handle error gracefully and there should be defined path for each error scenario for effective functioning of NF. With this enhanced functionality, Policy allows users to configure error codes.

The error codes are configured by adding customized values, for a defined condition, for the following fields:

- Error Description
- Diameter Result Code
- Experimental Result
  - Vendor ID
  - Experimental Result Code

Configure

To configure the diameter error codes for Policy, users can use any of the following ways:

- **CNC Console**: Perform the configurations on the Diameter Error Codes page. For more information, see Diameter Error Configurations .
- **REST API**: Perform the configurations using POST, PUT, or GET operations. For more information about REST API configuration, see *Cloud Native Core Policy REST Specification Document*.

Observe

The *occnp_diam_response_local _total* and *occnp_diam_response_network_total* metrics are applicable for this feature. For more details, see Diameter Gateway Metrics.

4.44 Configurations for Pre and Post Upgrade/Install Validations

This feature applies validation checks that are required on the application, databases, and its related tables before and after the upgrade/installation of Policy application.

On enabling this mandatory pre-flight and post-flight validation checks, for successful upgrade/ installation following are validated:

- does the related database exists
- does all the required tables exist
- does the required table schema exist for all the required tables
- does all the required infrastructure exists

This pre-flight and post-flight checks ensures that all the dependent databases, tables, schema, applications are in right order for performing successful update/installation.

For more information on how to how to set the parameter value for pre and post flight checks, see *Configurations for Pre and Post Upgrade/Install Validations* in *Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*.

4.45 Support Multiple Cluster Deployment at CNC Console

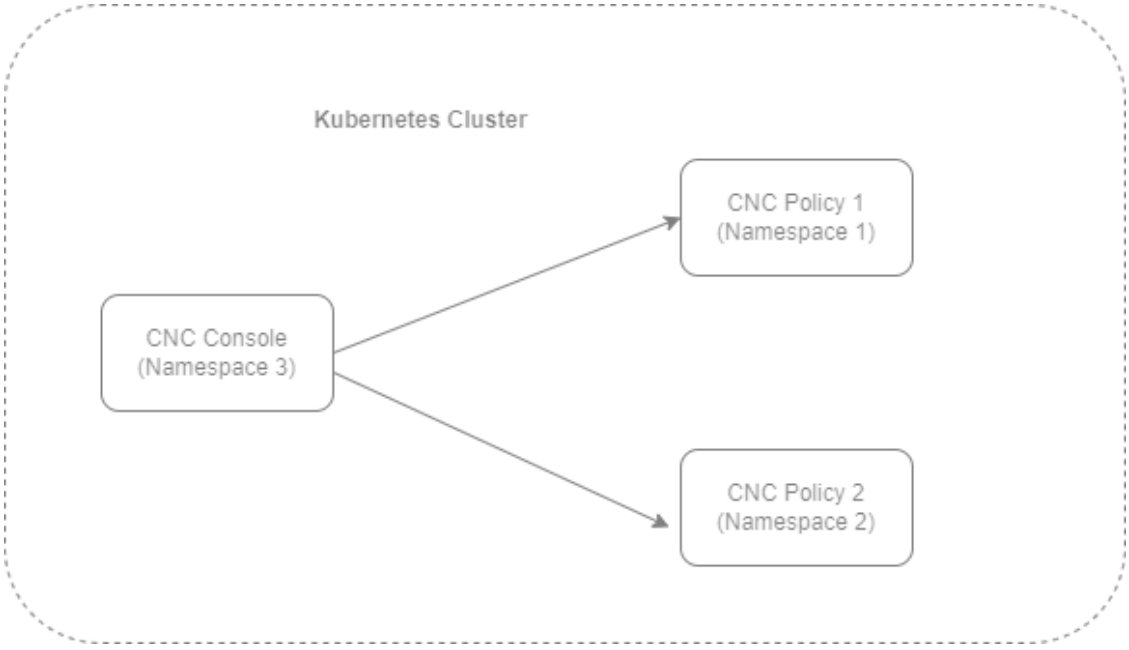
The CNC Console supports both single and multiple cluster deployments.

In a single cluster deployment, the CNC Console can manage NFs and Oracle Communications Cloud Native Core, Cloud Native Environment (CNE) common services deployed in the local Kubernetes clusters.

In a multiple instances deployment, the CNC Console can manage multiple Policy instances and CNE common services deployed within a Kubernetes cluster. For more information about single and multiple cluster deployments, see *Oracle Communications Cloud Native Core, Cloud Native Configuration Console Installation, Upgrade and Fault Recovery Guide*.

The following image represents a Kubernetes cluster with one instance of CNC Console and two instances of Policy. The single instance of the CNC Console is configuring two instances of Policy with different namespaces.

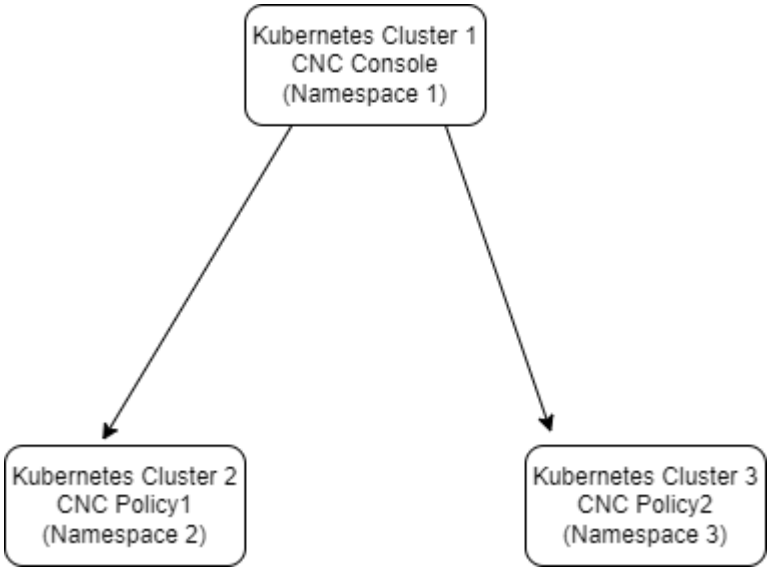
Figure 4-67 Support for Multiple Instance Deployment



With the support of multicluster deployment, Policy deployed in multiple Kubernetes clusters can be accessed using CNC Console. In a multicluster deployment, the CNC Console can manage Policy and CNE common services deployed in the remote Kubernetes clusters.

The following image represents multiple Kubernetes clusters with one CNC Console and two Policy deployments. The single instance of the CNC Console is configuring two instances of Policy with different namespaces deployed in different clusters.

Figure 4-68 Support for Multicluster Deployment



4.46 NetLoc Support

NetLoc is one of the 3GPP-defined features responsible for retrieving access network information in IMS network architecture. Depending on the operator's policy configuration and subscription, the NetLocfeature helps in fetching User Equipment (UE) time zone information and User location information from the access network. Such information is called Network Provided Location Information (NPLI). Operators can use the NPLI information for the following:

- Lawful interception
- Charging
- IMS Emergency Calls Routing
- Retention of Location Information Data
- Special Call Routing for Localized Services
- Location-based service triggering

Based on the response that it receives from SMF over the N7 interface, PCF provides this NPLI to the Application Function (AF) over the Rx interface.

The Application Function initiates a request towards PCF requesting NPLI using the "ACCESS_NETWORK_INFO_REPORT" within Specific-Action AVP and Required-Access-Info AVPs. PCF, upon receiving this request, initiates Npcf_SMPolicyControl_UpdateNotify procedure for AN_INFO event trigger requesting NPLI in terms of user location and/or user timezone information. SMF, upon receiving NPLI information from core network (AMF), provides that info PCF using Npcf_SMPolicyControl_Update procedure. Then, PCF forwards that information to AF using RAR message using 3GPP-User-Location-Info, 3GPP-MS-TimeZone, 3GPP-SGSN-MCC-MNC AVPs.

By default, this feature is not configured on the CNC Policy deployment. You must configure the NetLoc feature using the CNC Console or REST API.

Feature Negotiation

Once the feature has been enabled, feature negotiation needs to happen between AF and PCF during Npcf_PolicyAuthorization_Create Service operation. As defined by the 3GPP feature negotiation mechanism, the following conditions must be met for AF and PCF to agree upon the NetLoc feature:

- NF consumer or AF advertises the support for "NetLoc" feature within the attribute supportedFeatures (suppFeat) as part of SmPolicyData when sending a request to create SM policy association.
- In turn, PCF advertises the same value for the supportedFeatures (suppFeat) while sending the response for the policy association create request.

Managing NetLoc Support

Enable and Configure

By default, this feature is not configured on the CNC Policy deployment. You can opt to configure the NetLoc using the CNC Console or REST API.

On PCF Session Management page, under Service Configurations, select `NetLoc` from the drop-down menu of **Override Supported Features** parameter. For more information about the configurations, see PCF Session Management.

Using the REST APIs for Session Management Service, you can enable the feature by updating the value as `NetLoc` for the following parameter under the `system` group:

```
overrideSupportedFeatures": [
      "NetLoc"
    ],
```

Observe

No new alarms or alerts are introduced specific to this feature.

4.47 Subscription to Notification Support for Signaling Path Status

Policy supports notification of the AF signaling Transmission Path Status (SMF → Policy → AF), subscription to notification of the AF signaling Transmission Path Status (AF → Policy → SMF), and cancellation of subscription to notification of the AF signaling Transmission Path Status (AF → Policy → SMF).

PCRF-Core supports notification of the AF signaling Transmission Path Status (PGW → PCRF-Core → AF), subscription to notification of the AF signaling Transmission Path Status (AF → PCRF-Core → PGW), and cancellation of subscription to notification of the AF signaling Transmission Path Status (AF → PCRF-Core → PGW).

For Policy, during a PDU session establishment procedure, the SMF includes the `IMS_SIG` value within the `qosFlowUsage` attribute and the PCF accepts that default QoS flow is dedicated to IMS signaling, the PCF within the SmPolicyDecision data structure must include the `IMS_SIG` value within the `qosFlowUsage` attribute.

For more information on `IMS_SIG` value update in the CNC Console, see PCF Session Management.

For, PCRF-Core, during a PDU session establishment procedure, the SMF includes the `IMS_SIGNALLING` value within the `bearerusage` attribute and the PCF accepts that default QoS flow is dedicated to IMS signaling, the PCF within the SmPolicyDecision data structure must include the `IMS_SIGNALLING` value within the `bearerusage` attribute.

For more information on `IMS_SIGNALLING` value update in the CNC Console, see PCRF Core .

Notification of the AF Signaling Transmission

When the Policy is notified of the release of resources associated to the PCC or QoS rules corresponding with AF Signaling IP Flows, the Policy informs the AF about the release of the signaling Transmission path by sending a Re-Authorization Request (RAR) command to the AF.

The RAR includes the Specific-Action AVP set to the value "INDICATION_OF_RELEASE_OF_BEARER (4)" and the deactivated IP Flow encoded in the Flows AVP.

On Sm Update/CCR-U with the IMS Signalling Rule set as inactive, at a minimum, we are able to send an Rx ASR.

Figure 4-69 Notification of the AF Signaling Transmission in 5G

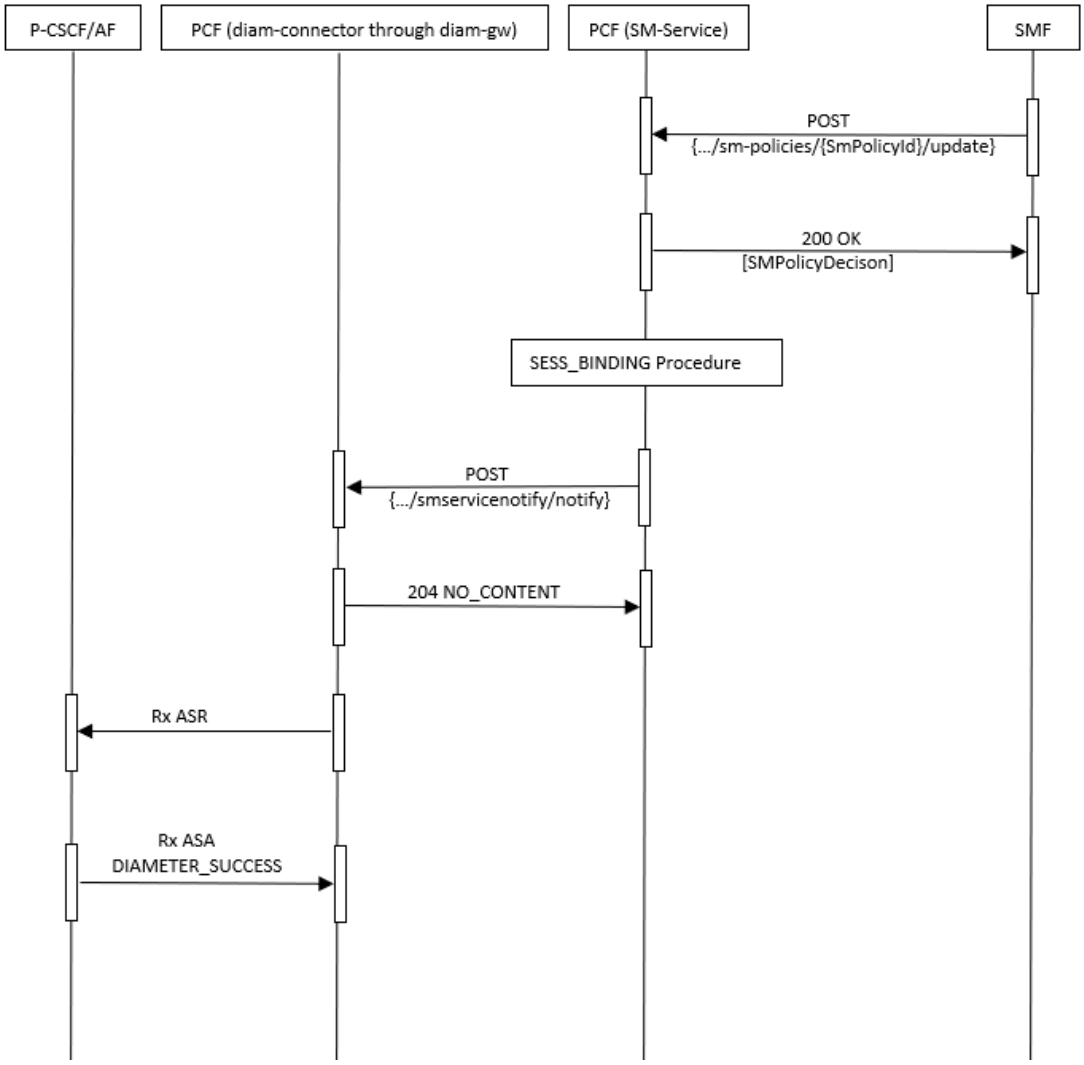
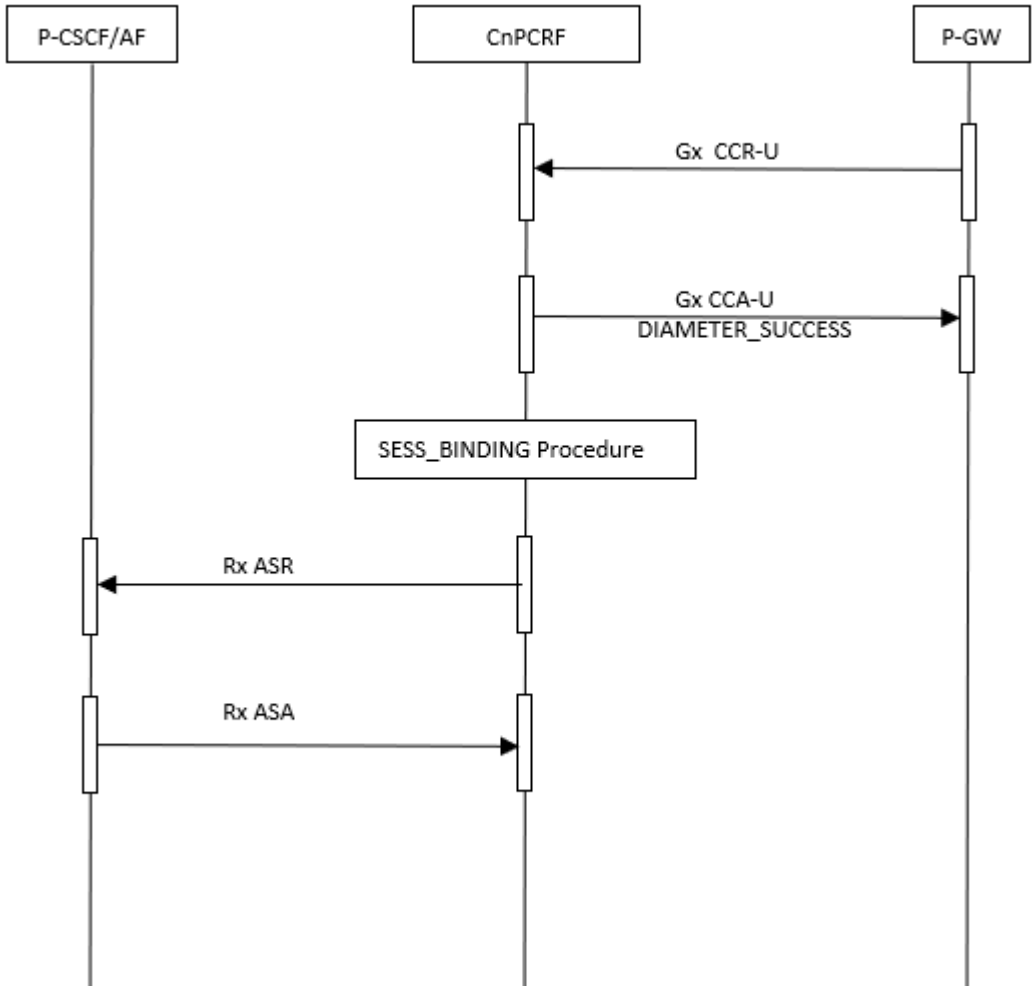


Figure 4-70 Notification of the AF Signaling Transmission in 4G



Subscription to notification of the AF Signaling Transmission

When AF receives an initial register SIP message from an attached UE, it subscribes to the notifications of the AF signaling transmission path status.

The AF provides the following:

- UE's IP address (using either the Framed-IP-Address AVP or the Framed-Ipv6-Prefix AVP)
- Specific-Action AVP requesting the subscription to "INDICATION_OF_RELEASE_OF_BEARER"
  - INDICATION_OF_RELEASE_OF_BEARER (4)

    In the AAR, this value indicates that the AF requests the server to provide a notification at the removal of a bearer
- The AF shall additionally provide a Media-Component-Description AVP including a single Media-Sub-Component AVP with the Flow-Usage AVP set to the value "AF_signaling"
- The Media-Component-Description AVP shall contain the Media-Component-Number AVP set to "0

On Rx AAR with the flow-usage as IMS Signalling, we link the Rx session with the IMS Signalling rule. We do not need to send a Gx RAR or SmUpdateNotify.

If we do not find an IMS Signalling rule, then bind it to the first PCC rule so we get an ASR only on CCR-T.

Figure 4-71 Subscription to notification of the AF Signaling Transmission in 5G

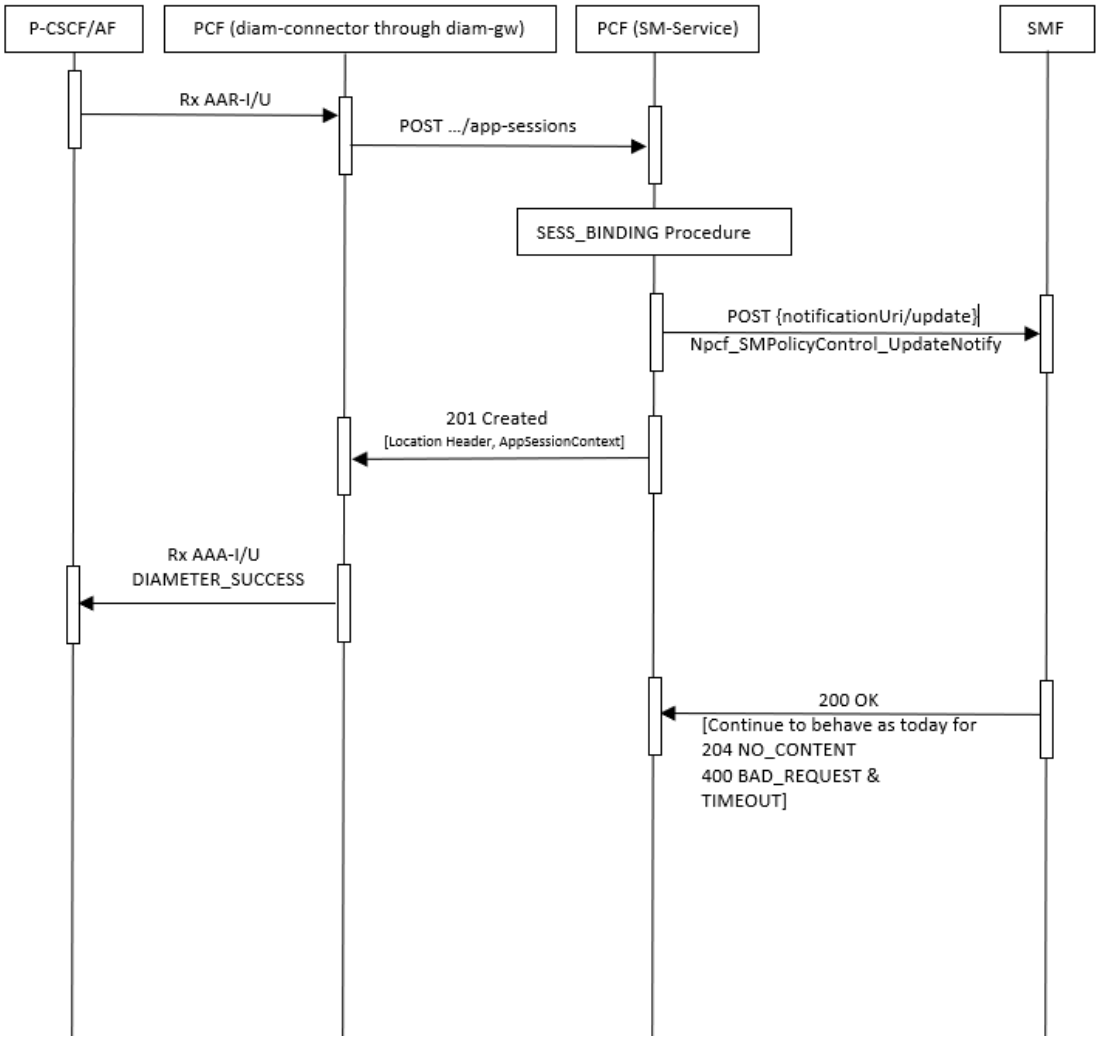
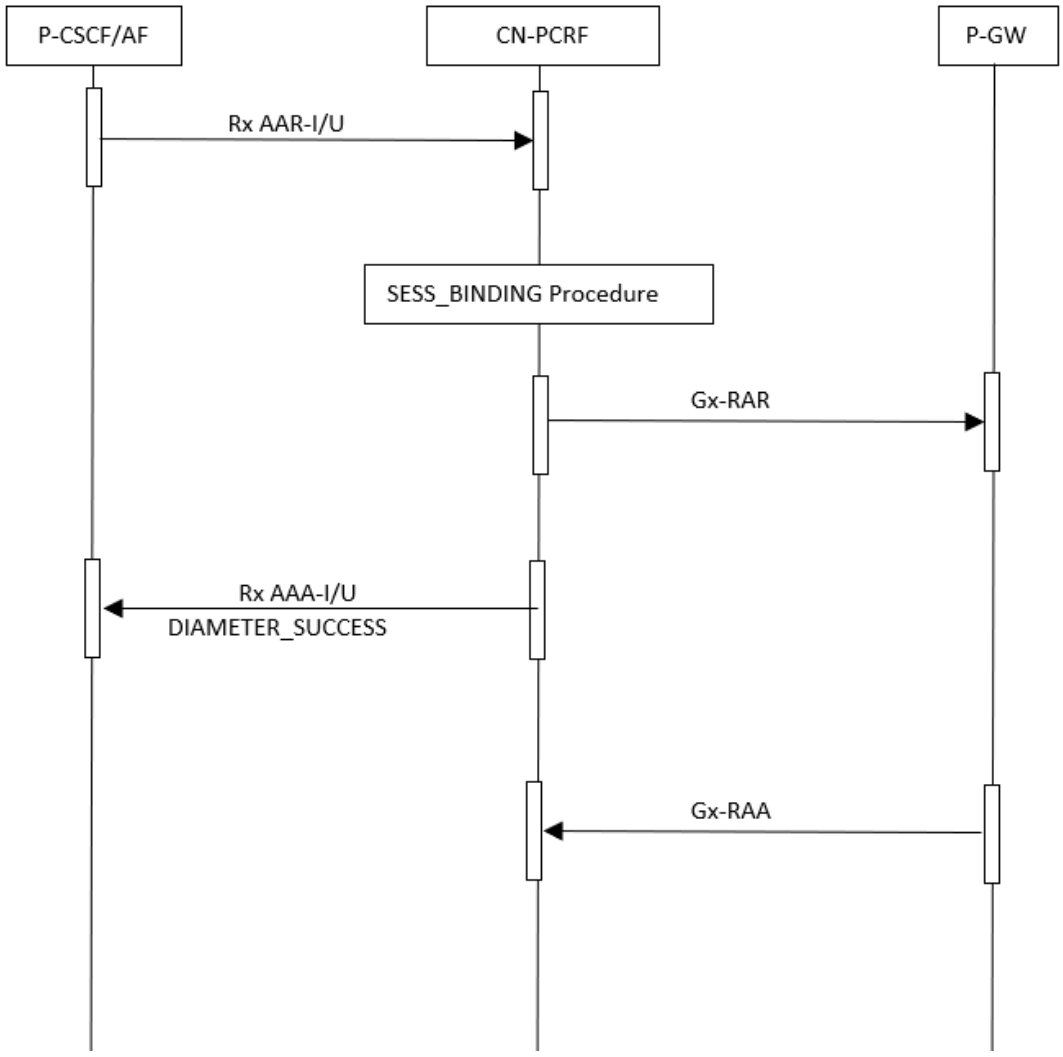


Figure 4-72 Subscription to notification of the AF Signaling Transmission in 4G



Cancellation of subscription to notification of the AF Signaling Transmission

If the Rx Diameter Session is only used for subscription to Notification of signaling Path Status, the AF may cancel the subscription to notifications of the status of the AF signaling transmission path. In this case, the AF uses a Session-Termination-Request (STR) command to the CNCPolicy, which gets acknowledged with a Session-Termination-Answer (STA) command.

On Rx STR, we simply unlink the Rx session from the Gx session / sm session and IMS Signalling Rule, but we do not remove the IMS Signalling Rule from the Gx or SM.

Figure 4-73 Cancellation of subscription to notification of the AF Signaling Transmission in 5G

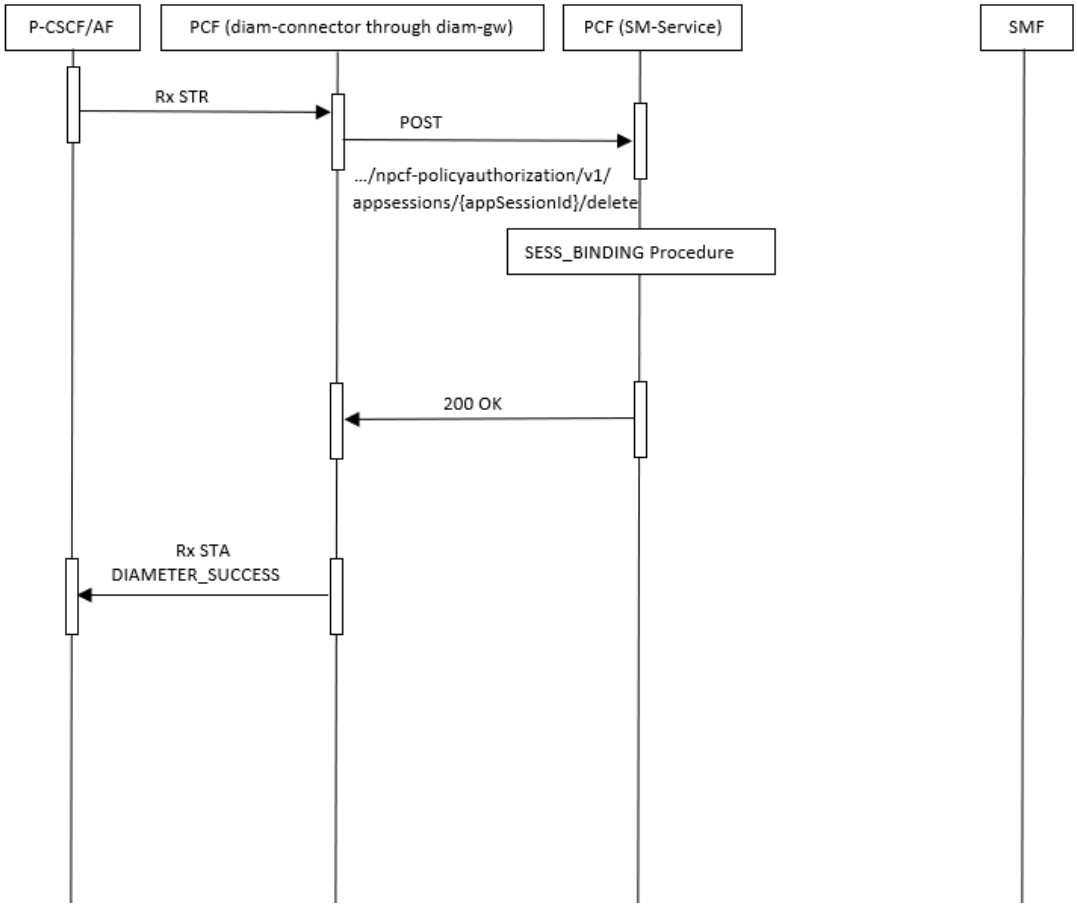
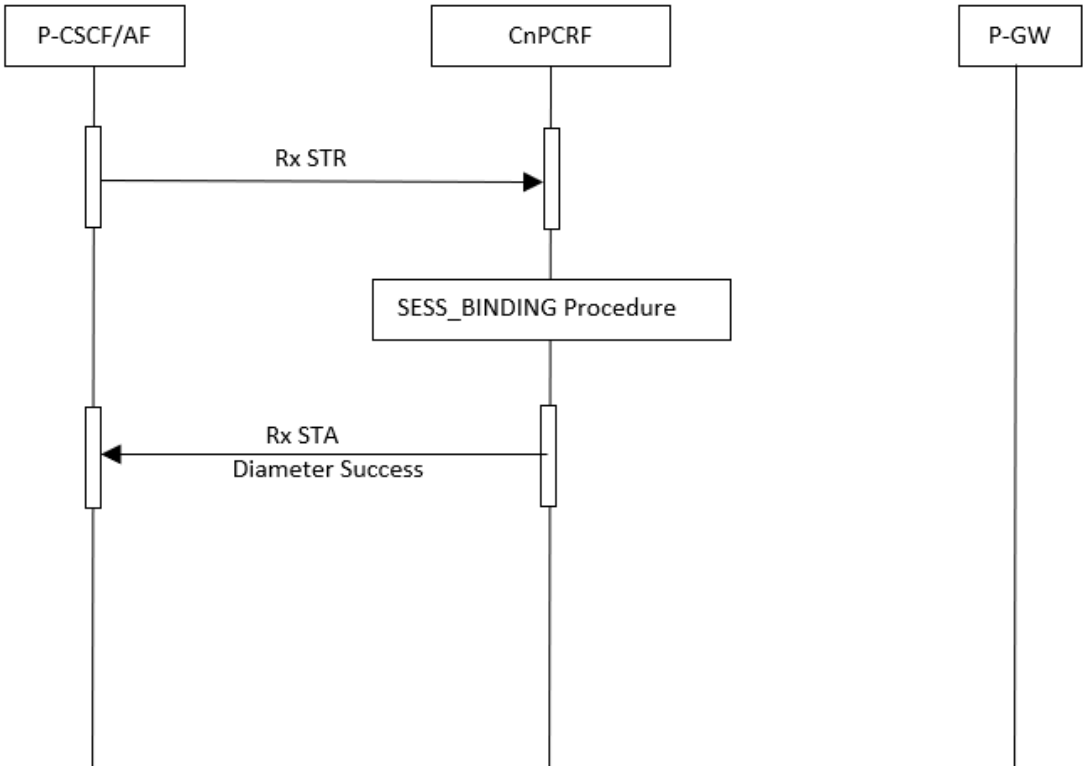


Figure 4-74 Cancellation of subscription to notification of the AF Signaling Transmission in 4G



4.48 Support for SessionRuleErrorHandling

The Session Rule Error Handling is a 3GPP 29.512 TS defined feature. If the SessionRuleErrorHandling feature is supported and SMF receives one or more session rules but it does not enforce some or all session rules then the SMF provides sessionRuleReports.

When SMF detects that the provisioning of some or all the session rules is unsuccessful, it sends a Session Rule Error Report in the `Npcf_SMPolicyControl_Update` and `Npcf_SMPolicyControl_UpdateNotify` Response. Depending on the Session Rule Error Report sent by SMF, the PCF may decide whether to retain the old session rule, re-installation, modification, or removal of the session rule.

As per the sessionRuleReport, you can write policy conditions matching the ruleStatus and sessRuleFailureCode received for the session rule and take action of installation, modification, and removal of the session rule. The blocks are available only when you select **PCC/Session Rule Error Report** under **PCF-SM** service while configuring the Policy Project. For more information about these blocks, see *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.

Feature Negotiation

Once the feature has been enabled, feature negotiation needs to happen between SMF and PCF during Npcf_SMPolicyControl_Create service operation. As defined by the 3GPP feature negotiation mechanism, the following conditions must be met for SMF and PCF to agree upon the SessionRuleErrorHandling feature:

- NF consumer or SMF advertises "SessionRuleErrorHandling" feature within the attribute supportedFeatures (suppFeat) as part of SmPolicyData when sending a request to create SM policy association.
- In turn, PCF advertises the same value for the supportedFeatures (suppFeat) while sending the response for the policy association create request.

Enable

By default, this feature is not enabled on the Policy deployment. You can enable the SessionRuleErrorHandling feature using the CNC Console or REST API.

- CNC Console: On PCF Session Management page, under Service Configurations, select `SessionRuleErrorHandling` from the drop-down menu of **Override Supported Features** parameter. For more information about the configurations, see PCF Session Management.
- REST API: Using the REST APIs for Session Management Service, you can enable the feature by updating the value as `SessionRuleErrorHandling` for the following parameter under the `System` group in the `{apiRoot}/oc-cnpolicyconfiguration/v1/services/pcfsm` REST API:

```
overrideSupportedFeatures": [
      "SessionRuleErrorHandling"
    ],
```

Observe

The following metrics have been enhanced for this feature:

- occnp_http_in_conn_request_total
- occnp_http_in_conn_response_total
- occnp_http_out_conn_request_total
- occnp_http_out_conn_response_total

> **Note**
>
> The "sessRuleReports" dimension has been added for *`occnp_http_in_conn_request_total`* and *occnp_http_in_conn_response_total* metrics.

The following metric have been added for this feature:

- occnp_sm_sess_rule_failure_total

For more information, see SM Service Metrics.

No new alerts are introduced for this feature.

Maintain

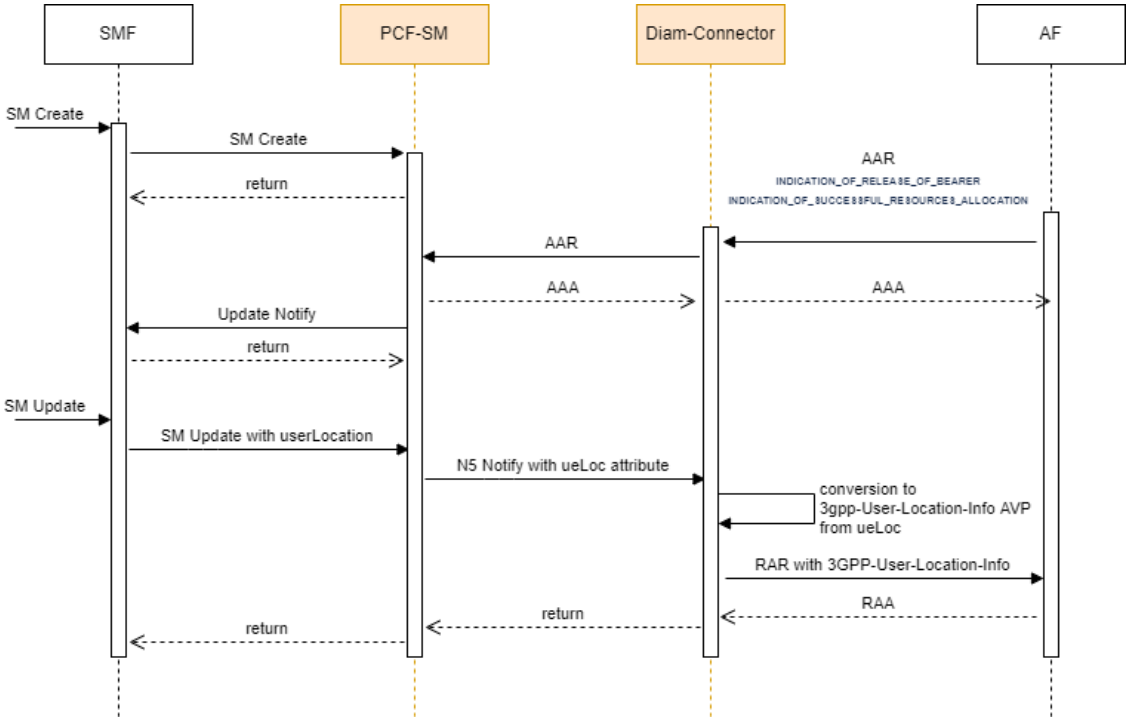
Logs are added for processing of sessionRuleReports attribute as part of SM-Update request and SM-UpdateNotify response.

4.49 3GPP-User-Location-Info AVP in Rx RAR

This section describes the inclusion of 3GPP-User-Location-Info in Rx RAR, when the UserLocation attribute is received on the SMF-N7 flows.

The following call flow describes the message flow from PCF to send 3GPP-User-Location-Info AVP towards AF:

Figure 4-75 3GPP-User-Location-Info Call Flow



PCF performs the following steps to send 3GPP-User-Location-Info AVP towards AF:

1. Application Function (AF) forwards the supported feature Attribute Value Pair (AVP) with LTE_Phase3 value in the first Authentication Request (AAR) of AF/Rx towards the PCF. In response, PCF exchanges the same AVP in Authentication Answer (AAA).
2. The AF subscribes for any of the specific-action between `INDICATION_OF_SUCCESSFUL_RESOURCE_ALLOCATION` and `INDICATION_OF_RELEASE_OF_BEARER`.
3. After establishing the AF session successfully, Session Management Function (SMF) sends `repPolicyCtrlReqTriggers` with `SUCC_RES_ALLO or RES_RELEASE` along with `ruleReports` and `userLocationInfo` to PCF.
4. SM service generates event notification request for Diameter Connector over N5 along with `ueLoc` attribute.
5. The Diameter Connector generates a Re-Auth-Request (RAR) with `3GPP-User-Location-Info` AVP for AF over Rx interface.

ueLoc Attribute

In case of 3GPP access, the PCF includes the user location information in the `ueLoc` attribute in N5 interface. The `ueLoc` atrribute has `eutraLocation` and `nrLocation` attributes.

`eutraLocation`: The data type for the `eutraLocation` attribute is `EutraLocation`. It has the following attributes:

Table 4-25 eutraLocation

Attribute Name	Data Type	Description
tai	Tai	The Type Allocation Code (TAC) of the Tracking Area Identity (TAI) is set to one reserved value. For an example 0x0000 is the TAC of a TAI attribute. If the TAI information is not available, see clause 19.4.2.3 of 3GPP TS 23.003 [7]).
ecgi	Ecgi	E-UTRA Cell Identity
ignoreEcgi	Boolean	This attribute indicates that the Ecgi is ignored. This attribute is set to `false` by default. When it is set as `true`, the Ecgi is ignored.
globalNgenbId	GlobalRanNodeId	It indicates the global identity of the ng-eNodeB in which the UE is currently located.
globalENbId	GlobalRanNodeId	It indicates the global identity of the eNodeB in which the UE is currently located.

`nrLocation`: The data type for the `nrLocation` attribute is `NrLocation`. It has the following attributes:

Table 4-26 NrLocation

Attribute Name	Data Type	Description
tai	Tai	The TAC of the TAI is set to one reserved value. For an example 0x0000 is the TAC of a TAI attribute.
ncgi	Ncgi	NR Cell Identity
ignoreNcgi	Boolean	This attribute indicates that the Ncgi is ignored. This attribute is set to `false` by default.
globalGnbId	GlobalRanNodeId	This attribute indicates the global identity of the `gNodeB` in which the UE is currently located.

For more information on ueLocation, see the 3GPP technical specification 29.514.

In the RAR message, `GeographicLocation` is used. Following are the types of GeographicLocation used in the RAR message:

- TAI(128): Tracking Area ID for E-UTRAN.
- ECGI(129): Evolved Cell Global ID for E-UTRAN.

- TAI_ECGI(130): Tracking Area ID and Evolved Cell Global ID for E-UTRAN.
- ENODEB(131): global identity of the ng-eNodeB for E-UTRAN.
- TAI_ENODEB(132) : Tracking Area ID and global identity of the eNodeB for E-UTRAN.
- EXTENDEDNODEB(133): global identity of the eNodeB for E-UTRAN.
- TAI_EXTENDEDNODEB(134) : Tracking Area ID and global identity of the eNodeB for E-UTRAN.
- NCGI(135) : NR Cell Global Identity for NR.
- TAI5G(136) : Tracking Area ID for NR.
- TAI5G_NCGI(137) : Tracking Area ID and NR Cell Global Identity for E-UTRAN.
- NGRANNODE(138) : global identity of the gNodeB for NR.
- TAI5G_NGRANNODE(139) : Tracking Area ID and global identity of the gNodeB for NR.

Following is the sample N7 update request for the userLocationInfo attribute:

```
"userLocationInfo": {
    "eutraLocation": {
      "tai": {
        "plmnId": {
          "mnc": "313",
          "mcc": "350"
        },
        "tac": "790"
      },
      "ecgi": {
        "plmnId": {
          "mnc": "313",
          "mcc": "350"
        },
        "eutraCellId": "AB0912"
      },
      "ageOfLocationInformation": 233,
      "ueLocationTimestamp": "2019-03-13T06:44:14.34Z",
      "geographicalInformation": "AAD1234567890123",
      "geodeticInformation": "AAD1234567890123BCEF",
      "globalNgenbId": {
        "plmnId": {
          "mnc": "313",
          "mcc": "350"
        },
        "n3IwfId": "n3iwfid"
      }
    }
  }
```

Following is the sample N5 Notification Request for the ueLoc attribute:

```
"ueLoc": {
    "eutraLocation": {
        "tai": {
        "plmnId": {
          "mcc": "313",
```

```
          "mnc": "350"
        },
        "tac": "790"
      },
      "ecgi": {
        "plmnId": {
          "mcc": "313",
          "mnc": "350"
        },
        "eutraCellId": "AB0912"
      }
    },
    "nrLocation": null,
    "n3gaLocation": null
  }
```

Following is the sample Diameter RAR message:

```
Sample Diameter RAR request:
Diameter Message: RAR
Version: 1
Msg Length: 232
Cmd Flags: REQ,PXY
Cmd Code: 258
App-Id: 16777236
Hop-By-Hop-Id: 4144052655
End-To-End-Id: 2394693330
  Session-Id (263,M,l=9) = 1
  Origin-Host (264,M,l=28) = diam-conn.oracle.com
  Origin-Realm (296,M,l=18) = oracle.com
  Destination-Realm (283,M,l=24) = test.example.com
  Destination-Host (293,M,l=28) = diamcliaf.oracle.com
  Auth-Application-Id (258,M,l=12) = 16777236
  Specific-Action (513,VM,v=10415,l=16) =
INDICATION_OF_SUCCESSFUL_RESOURCES_ALLOCATION (8)
  Flows (510,VM,v=10415,l=44) =
    Media-Component-Number (518,VM,v=10415,l=16) = 1
    Flow-Number (509,VM,v=10415,l=16) = 2
  User-Location-Info-3GPP (22,V,v=10415,l=25) = Type=TAI_ECGI(130)
MCCMNC=313350 TAC=1936 ECI=11208978
```

For more information, see the following 3GPP specification:

- 29.512
- 29.514
- 29.214
- 29.229

Enable

This feature is enabled automatically at the time of Policy installation.

Configure

Diameter connector has an environment variable `PRIORITY_SPECIFIC_ACTION` which prioritizes the action while sending RAR message, if AF has subscribed for `INDICATION_OF_FAILED_RESOURCE_ALLOCATION` and `INDICATION_OF_RELEASE_OF_BEARER`.

4.50 Support for Server Header

PCF handles various requests from consumer Network Functions (NFs) and other network entities over HTTP protocol. On receiving these requests, PCF validates and processes them before responding to these requests. In case, PCF sends an error response, then the consumer NFs need to know the source of the error to trouble shoot the error and take corrective measures. The integration of this feature at PCF helps to determine the originator of the error response.

This feature offers the support for Server Header in PCF responses, which contains information about the origin of an error response and the type of the error encountered. The Server Header includes the type of NF as "NF Type", followed by a “-” and the identity of the NF or the network entity. It is expected to be present in all PCF responses in the following format:

<NF_Type>-<NF_Instance_Id>

Where,

- `<NF Type>` is the type of the NF generating the error.
- `<NF Instance-Id>` is the unique identifier of the NF instance generating the error response.

For example: PCF-54804518-4191-46b3-955c-ac631f953ed8

The inclusion of the Server header in the PCF response is configurable, and can be enabled or disabled using a flag. Also the error codes that are included as part of the Server header in the error response are also configurable. The configuration of these parameters are done through either with REST APIs that are exposed through configuration server or Helm Configurations.

The operation mode that is either `REST` or `HELM` for Server Header configuration is done using the below flag:

```
ingress-gateway:
  serverHeaderConfigMode: REST # Possible values: HELM, REST. Based on this
value, the feature flag for "server"  header will need to be enabled either
in Helm configuration or Rest  configuration.
```

> **Note**
>
> Nf Type and Nf Instance Id are mandatory fields for Server Header to get included in the error response. If either of the fields Nf Type or Nf Instance Id are configured as empty, then the Server Header will not get included in the error response.

Managing Server Header

Enable

By default, this feature is disabled.

You can enable the Server Header feature using Helm or REST API configurations as follows:

- **Helm:** To enable the server header feature using Helm configuration, set the value for parameter `serverHeaderConfigMode` to `HELM` in the `custom-values.yaml` file. Then, set the value for parameter `serverHeaderEnabled.enabled` to `true` under routesConfig for `ingress-gateway`.
  For more information, see the *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.
- **REST API:** To enable the server header feature using REST configuration, set the value for parameter `serverHeaderConfigMode` to `REST` in the `custom-values.yaml` file. Using REST API, set the `enabled` parameter to `true` in the following resource URI:

  {apiRoot}/PCF/nf-common-component/v1/igw/serverheaderdetails

  For more information, see the section Server Header at Ingress Gateway in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

You can configure the server header feature using the REST API or CNC Console:

- **Configure using REST API:**Perform the REST API configurations in the following sequence to configure this feature:
  1. Configure **serverheaderdetails** to enable the feature.
     {apiRoot}/PCF/nf-common-component/v1/igw/serverheaderdetails
  2. Configure **routesconfiguration** to map route ID and its corresponding route-level configuration.
     {apiRoot}/PCF/nf-common-component/v1/igw/routesconfiguration
  3. Configure **errorcodeserieslist** to update the **errorcodeserieslist** that are used to list the configurable exception or error for an error scenario in Ingress Gateway.
     {apiRoot}/PCF/nf-common-component/v1/{serviceName}/errorcodeserieslist

  > **Note**
  >
  > If you define server header configuration at both global and route levels, the route level configuration takes precedence over the global level configuration.

  For more information, see the "Server Header at Ingress Gateway" section in *Oracle Communications Cloud Native Core, Converged Policy REST API Specification Guide*.
- **Configure using helm:** When parameter `serverHeaderConfigMode` is set to `HELM` and to configure the Server Header at Ingress Gateway, you need to perform the helm configurations either at Global or at Route level.

```
# All attributes under "serverHeaderDetails" will need to be configured
only if "serverHeaderConfigMode" is set as "HELM"
serverHeaderDetails:
  enabled: true
  errorCodeSeriesId: E1
  configuration:
    nfType: PCF
    nfInstanceId: INS-1

# Use below configuration to define errorCodeSeries list
```

```
errorCodeSeriesList:
  # Value of "id" attribute will need to used for assigning
"errorCodeSeriesId" either at Global or Route level conf for Server header.
- id: E1
  errorCodeSeries:
  - errorSet: 4xx
    errorCodes:
    - 400
    - 408
  - errorSet: 5xx
    errorCodes:
    - 500
    - 503
- id: E2
  errorCodeSeries:
  - errorSet: 4xx
    errorCodes:
    - -1
```

Following Helm Configuration performed at Route Level:

```
routesConfig:
- id: backend_ms1_route
  uri: https://backend-ms1:8440/
  path: /ms1/**
  order: 1
  metadata:
    # All attributes under "serverHeaderDetails" will need to be
configured only if "serverHeaderConfigMode" is set as "HELM" and Route
level configuration is required. If not defined, Global configurations
will be used
    serverHeaderDetails:
      enabled: true     # Since this flag is set to true at Route level,
"server" header configuration will be enabled for this Route with
respective "errorCodeSeriesId" as E2
      errorCodeSeriesId: E2 # This attribute will need to be defined if
"server" header configuration is enabled at Route level.
- id: backend_ms2_route
  uri: https://backend-ms2:8550/
  path: /ms2/**
  order: 2
  metadata:
    # All attributes under "serverHeaderDetails" will need to be
configured only if "serverHeaderConfigMode" is set as "HELM" and Route
level configuration is required. If not defined, Global configurations
will be used
    serverHeaderDetails:
      enabled: false    # Since this flag is set to false at Route level,
"server" header configuration will be disabled for this Route altogether.
```

> **Note**
>
> If you define server header configuration at both global and route levels, the route level configuration takes precedence over the global level configuration.

For more information, see the *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.

- **Configure using CNC Console:** A new group, **NF Server Settings** is added to the NF Communication Profile page. For more information, see NF Communication Profiles.

4.51 SBI Timer Handling

PCF-SM service receives requests from ingress Gateway with the help of the following 3GPP headers:

- `3gpp-Sbi-Origination-Timestamp`- contains the date and time (with a millisecond granularity) when the originating entity initiated the request.
- `3gpp-Sbi-Sender-Timestamp`- contains the date and time (with a millisecond granularity) at which an HTTP request or response is originated.
- `3gpp-Sbi-Max-Rsp-Time`- contains the time duration expressed in milliseconds since the absolute time is indicated in the 3gpp-Sbi-Origination-Timestamp header.

The timer profile sets request timeout values for UDR, CHF, BSF, SMF, and AMF messages in applications or interfaces. Policy applies the specific timeout profile while sending requests to external NFs. For more information on managing timer configurations, see Support for Timer Configurations.

The timeout value used towards different interfaces of PCF such as UDR, CHF, BSF, SMF, and AMF also depends on the message communication towards the HTTP server. The message communication towards the HTTP server can be synchronous or asynchronous:

1. Synchronous:
   - When Timer headers are available:
     - When current time is subtracted from the sum of `3gpp-Sbi-Origination-Timestamp` and `3gpp-Sbi-Max-Rsp-Time` and the result is greater than zero, then it is considered as `potential_requestTimeout`. In this case, if the validation fails you must immediately fail the request towards upstream. Thereafter, a similar check is done by the upstream service.
     - If TimerProfile is configured and applicable for a respective interface on service, then the minimum value out of `potential_requestTimeout` or `TimerProfile_value` is taken as request timeout.
     - If TimerProfile is not configured or applicable for respective interfaces, then the minimum value out of `potential_requestTimeout` or `Static timeout value of service` is taken as request timeout. Here, the static timeout value is the value timeout that the user configures in the Helm.
   - When Timer headers are not available:
     - If TimerProfile is configured and applicable for respective interfaces of the service, then `TimerProfile_value` is used as request timeout.
     - If TimerProfile is not configured or applicable for respective interfaces of the service, then static timeout is used as request timeout.

2. Asynchronous
    - If TimerProfile is configured for a respective interfaces on service, then use value of the `TimerProfile` request timeout.
    - If TimerProfile is not configured or applicable for respective interface on service, then set the static timeout as request timeout. While setting the static timeout value, you must ensure that the value of static timeout is greater than any timeout profile value.

The timeout value is calculated as per message instead of per interface. It allows the user to configure a different timeout, whether the UDR connector or NRF interface is sending a Get or Subscribe message. The user can calculate the timeout using UDR messages. For more information on managing the timeout, see Support for Timer Configurations.

Managing SBI Timer

Enable

To enable the SBI Timer Header, set the `isSbiTimerEnabled` parameter to true, under the ingress-gateway configurations in the custom values.yaml file for Policy.

If the `isSbiTimerEnabled` parameter is 'true' then:

- `3gpp-Sbi-Sender-Timestamp`, `3gpp-Sbi-Max-Rsp-Time`, and `3gpp-Sbi-Origination-Timestamp` are used along with route level (if configured) and global level request timeout to calculate final request timeout.
- After calculating the final request timeout, the original values of `3gpp-Sbi-Sender-Timestamp`, `3gpp-Sbi-Max-Rsp-Time`, and `3gpp-Sbi-Origination-Timestamp` are published in the `Orig-3gpp-Sbi-Sender-Timestamp`, `Orig-3gpp-Sbi-Max-Rsp-Time`, and `Orig-3gpp-Sbi-Origination-Timestamp` respective custom headers.

If `isSbiTimerEnabled` is 'false', then the SBI headers are not taken into consideration even if, they are present and the custom headers are not published.

Configure

When `isSbiTimerEnabled` flag is true in Ingress Gateway and publish headers flag is enabled, Egress Gateway needs to get the original value from the corresponding orig header and set it to the 3gpp header. This can be done by adding an entry filter per route in egress gateway section of custom-values.yaml file.

Following is an example of how to configure the entry filter to ensure that the '3gpp-Sbi-Origination-Timestamp' header retains the original value stored in the 'Orig-3gpp-Sbi-Origination-Timestamp' header for the UDR route:

```
routesConfig:
    - id: udr_direct
      uri: http://dummy.dontchange2
      path: /nudr-dr/**
      order: 3
      metadata:
        # Configuration done at route level will take precedence over same
configuration done at global level
        httpsTargetOnly: false
        httpRuriOnly: false
        sbiRoutingEnabled: false
        sbiRoutingWeightBasedEnabled: false
        configurableErrorCodes:
```

```
      enabled: false
      #errorProfileName are always defined at global level
      errorScenarios:
        - exceptionType: "VIRTUAL_HOST_RESOLUTION_ERROR"
          errorProfileName: "ERR_114"
        - exceptionType: "INVALID_OAUTH_TOKEN_REQUEST"
          errorProfileName: "ERR_115"
        - exceptionType: "OAUTH_INTERNAL_ERROR"
          errorProfileName: "ERR_115"
        - exceptionType: "OAUTH_TOKEN_RETRIEVAL_FAILURE"
          errorProfileName: "ERR_115"
        - exceptionType: "OAUTH_NRF_RESPONSE_FAILURE"
          errorProfileName: "ERR_115"
        - exceptionType: "CONNECTION_TIMEOUT"
          errorProfileName: "ERR_100"
        - exceptionType: "REQUEST_TIMEOUT"
          errorProfileName: "ERR_200"
  filterNameReqEntry:
    name: CustomReqHeaderEntryFilter
    args:
      headers:
        - methods:
            - ALL
          headersList:
            - headerName: 3gpp-Sbi-Origination-Timestamp
              defaultVal: <default_value>
              source: incomingReq
              sourceHeader: Orig-3gpp-Sbi-Origination-Timestamp
```

You can configure the parameters for SBI Timer by updating the custom-values.yaml file for Policy. For more information about configuring the parameter value, see *Late Arrival Handling Configuration* in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*.

Observe

The `ocpm_udr_tracking_request_timeout_total`, `ocpm_udr_tracking_request_timeout_total`, `ocpm_egress_request_timeout_total`, and `ocpm_timeout_total` metrics are applicable for SBI timers. For information related to SBI Timer metrics, see CNC Policy Metrics .

4.52 Detection and Handling of Late Arrival Requests

PCF Service receives requests from Ingress Gateway with the 3GPP headers. These requests help in the detection of the response time for the SM service. Following headers are received from the Ingress Gateway:

- `3gpp-Sbi-Origination-Timestamp`- It contains the timestamp when the originating entity initiates the request.
- `3gpp-Sbi-Max-Rsp-Time`- It contains the time duration expressed in milliseconds since the absolute time is indicated in the 3gpp-Sbi-Origination-Timestamp header.
-

- `3gpp-Sbi-Sender-Timestamp`- It contains the date and time (with a millisecond granularity) at which an HTTP request or response is originated from the previous NF.

PCF must be able to read the `3gpp-Sbi-Origination-Timestamp` and `3gpp-Sbi-Max-Rsp-Time` headers to be able to respond as per the request. The following scenarios can be considered:

- If the sum of `3gpp-Sbi-Origination-Timestamp` and 3gpp-Sbi-Max-Rsp-Time is less than the current time, the PCF service must reject the message with a 504 HTTP code and the message: "TIMED_OUT_REQUEST".
- If the request does not include either `3gpp-Sbi-Origination-Timestamp`, `3gpp-Sbi-Sender-Timestamp`, or `3gpp-Sbi-Max-Rsp-Time` headers, then this request is accepted and the call flow continues as normal with the inclusion of collision detection.
- If `3gpp-Sbi-Max-Rsp-Time` receives a negative value, this header is considered invalid. In this case, the service fallbacks to the default behavior and accepts the request irrespective of the origination/Sender timestamp value.

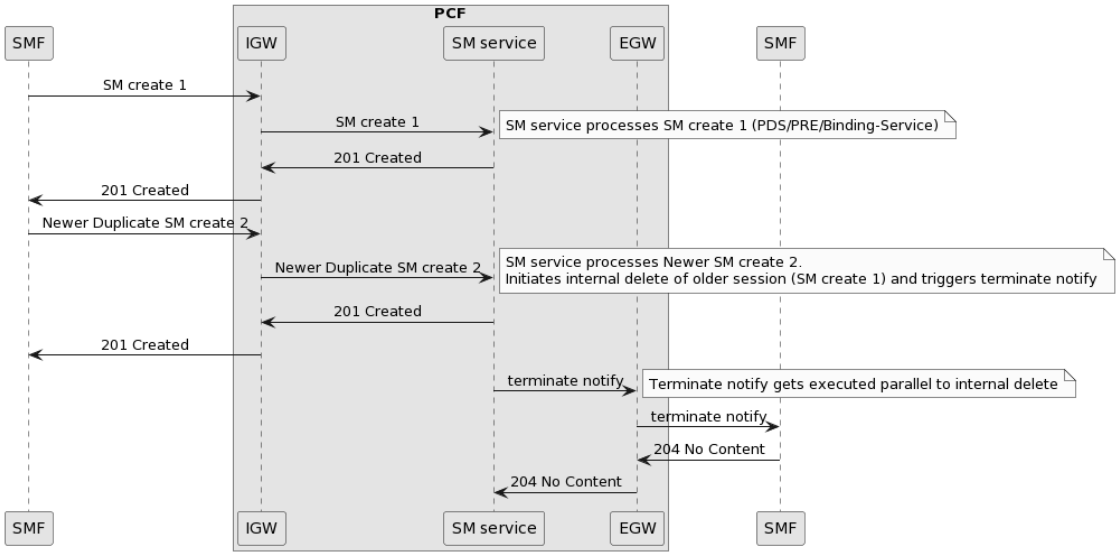
In case of collision detection, PCF needs to trigger a cleanup internally on the rest of the services so that there is only one association per user at a time.

Collision Detection of SM PDU Sessions

When SM Service receives a new SM PDU Session Create request and if the value of `SYSTEM.COLLISION_DETECTION.TERMINATE_NOTIFY.ENABLED` advanced settings key is set to `true`, SM Service detects and cleans up duplicate requests of SM Sessions (Collision Detection) and sends Terminate Notify to SMF when .

SM service deletes the older session from SMPolicyAssociation database and sends the Terminate Notify request to SMF.

Figure 4-76 Sample call flow for sending Terminate Notify when duplicate session is found (collision detection) and SM service cleans up the older session from database



1. SM service receives a SM CREATE request from SMF through Ingress Gateway.
2. SM service sends a GET request to PDS to get the subscriber details from UDR, initiates a binding session request with Binding Service and then sends a request to PRE for policy

evaluation. After the session is successfully created, SM services responds to SMF with 201 created message.

3. SM service receives another CREATE request from SMF, which is a duplicate of the session that is already created.
4. SM service processes the new CREATE request and deletes the older session.
5. SM service sends a Terminate Notify message to SMF through Egress Gateway.
6. After deleting the details of the older session, SMF responds to SM service with 204 No Content message.

Managing Late Arrival requests and Collision Detection

Enable

For collision detection, the **SYSTEM.COLLISION.DETECTION** parameter is available to the user. It is set to false, by default. When the parameter is set to true, PCF-SM checks for collision in the SMPolicyAssociation table, based on User Equipment information (ie. SUPI) and the PDU Session ID.

PCF-UE and PCF-AM checks for collision in the UEPolicyAssociation and AMPolicyAssociation table respectively, based on User Equipment information (ie. SUPI).

Configure

You can configure the parameters for late arrival handling by updating the custom values file for Policy. For more information about configuring the parameter value, see *Late Arrival Handling Configuration* in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.

Observe

Policy provides metrics specific to late arrival requests and collision detection. For information related to late arrival requests and collision detection metrics, see Late Arrival Requests and Collision Detection Metrics .

4.52.1 PCF Support for Detection and Handling of Late Arrival Requests in BSF

Each microservice propagates the 3gpp-Sbi-Origination-Timestamp header and updates the 3gpp-Sbi-Sender-Timestamp from PCF to BSF with the current timestamp when it receives a request from SMF or SCP.

For adding Detection and Handling of Late Arrival functionality in BSF, it is necessary to make sure these headers and a new Custom header are propagated from PCF to BSF. Moreover, as required by the user, PCF supports to set the value of Custom-Sbi-Sender-Timestamp header.

To create Custom-Sbi-Sender-Timestamp it is necessary to add the following configuration to PCF ingress-gateway:

```
routesConfig:
    - id: sm_create_session_route
      uri: http://{{ .Release.Name }}-occnp-pcf-sm:
{{ .Values.global.servicePorts.pcfSmServiceHttp }}
      path: /npcf-smpolicycontrol/*/sm-policies
      order: 1
      method: POST
      readBodyForLog: true
```

```
        filters:
          subLog: true,CREATE,SM
          customReqHeaderEntryFilter:
            headers:
              - methods:
                - POST
                headersList:
                  - headerName: 3gpp-Sbi-Message-Priority
                    defaultVal: 24
                    source: incomingReq
                    sourceHeader: 3gpp-Sbi-Message-Priority
                    override: false
                  - headerName: Custom-Sbi-Sender-Timestamp
                    defaultVal: func:currentTime(EEE, d MMM yyyy HH:mm:ss.SSS
z,gmt)
                    source: incomingReq
                    sourceHeader: 3gpp-Sbi-Sender-Timestamp
                    override: false
```

4.53 Support for Session Retry and Alternate Route Service

For previous releases, Policy and PCF were configured with primary and secondary NRF or SCP statically by limiting to a specific number. Starting with release 1.8.0, Policy provides the Session Retry functionality. This feature enables the alternate recovery mechanisms to mitigate the impact of any unavailable resource. The policy system provides flexible and configurable retry behavior for selected interfaces.

A retry profile specifies when and how the signaling message from one network function to another are retried and rerouted on failures. For example, if heartbeat messages that PCF sends to NRF fail consistently, the PCF should choose a different (secondary or next priority) NRF that is available to send the subsequent heartbeat messages. Similarly, the retry or reroute mechanisms are required for messages going towards UDR, CHF, and so on. Even the notifications to SMF and AMF may need to be retried and rerouted based on the configuration.

Policy supports the session retry functionality for failed notifications. In case of a notification towards SMF fails, then the PCF sends an On-Demand discovery towards NRF based on the SetID received in the CREATE or UPDATE request from the SMF as part of the *3gpp-Sbi-Binding* header. NRF performs a SetID based resolution and sends back a list of NFs under the same SetID. PCF selects the alternate SMF based on the configurations and sends the next notification.

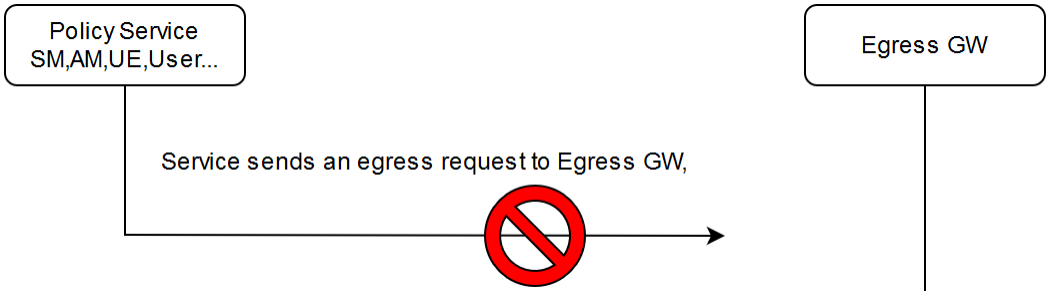
> **Note**
>
> The following conditions must be met for the Session Retry for Notifications functionality: :
>
> - The **NF Set Resolution** configuration must be enabled. For more information about the configuration, see Managing Session Retry and Alternate Route Service.
> - NF must send the Binding headers with `nfSetid` in the CREATE or UPDATE request.

Retry

Retry is initiated when PCF microservices initiate an egress request, but it fails to reach the egress gateway. The reason can be connection failure or Aspen Service Mesh failure. Session retry enables alternate recovery mechanisms to mitigate impact of any unavailable resource. Policy system provides flexible and configurable retry behavior for selected interfaces. You can configure retry profile for the following destination NFs: UDR, CHF, BSF, SMF, AMF, and AF through CNC Console.

The following figure shows a scenario when PCF microservice initiates an egress request, but it fails to reach the egress gateway:

Figure 4-77 PCF Microservice Initiates Egress Request

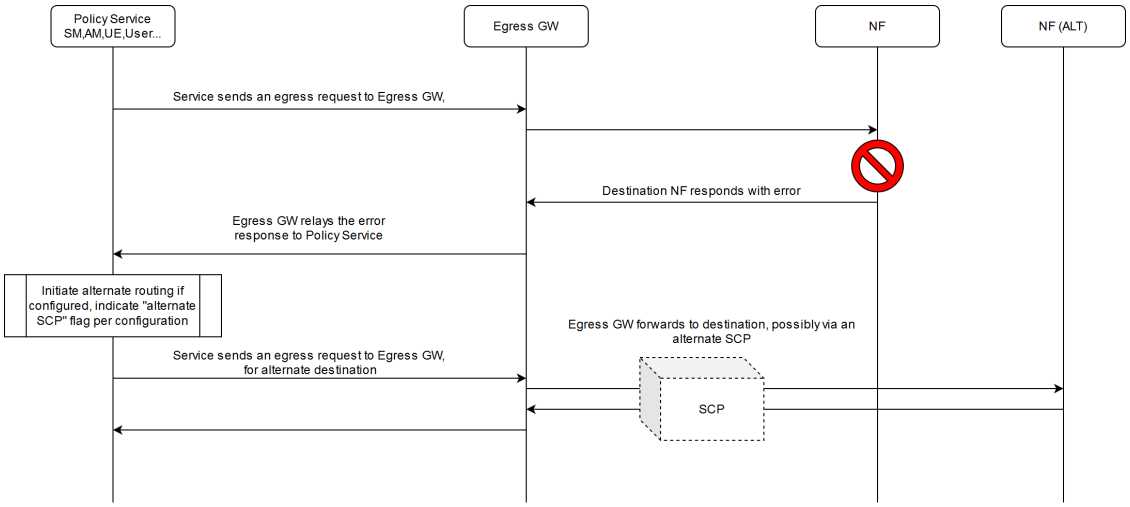


Alternate routing

Alternate routing is initiated when egress gateway fails to reach external network function (NF) or external NFs fail with an error code, or external NFs do not respond. Alternate destination can be chosen by using static configuration or by using DNS-SRV records. For more information about static and DNS-SRV configurations, see "Configurable Parameters for Alternate Route Service Configuration" in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*.

The following figure shows a scenario when egress gateway reaches the NF, but NF responds with an error code:

Figure 4-78 Egress Gateway get Error Response



To support this behavior, PCF provides the capability to perform DNS SRV Query. The DNS SRV Query discovers the NFs, such as NRF, SCP, and alternate producers. It also notifies consumers during failures to enable the retry mechanisms. This feature provides support to configure virtual FQDNs and adds capability to perform DNS SRV or local configuration Look up to retrieve alternate failover NFs which can be maintained dynamically at the DNS Server.

With SRV Records, you can configure and maintain NF FQDN dynamically at the DNS Server, which can be further selected by Policy, when there is a NF failure. This is achieved by performing a SRV query on the virtual FQDN configured at the Policy, rather than configuring primary and secondary NRF or SCP, statically in every Policy, only during instantiation time. This option of DNS lookup for SRV records would also provide alternate NFs to the Policy during failover.

Egress Gateway

Policy supports session retry functionality at Egress Gateway. When Egress Gateway receives HTTP error response based on the matching `errorcriteria`, there is retry of failed request through an alternate SCP based on the defined `erroractionset`. The `errorcriteria` defines matching HTTP methods, status codes, and causes (optional). The `erroractionset` defines number of reattempted counts and blacklist configurations. The respective `errorcriteria` and `erroractionset` should be mapped to the `routesConfig`.

The following sample shows the mapping of `errorcriteria` and `erroractionset` with `routesConfig`:

```
routesConfig:
 - id: scp_direct2
   uri: https://dummy.dontchange2
   path: /<Intended Path>/**
   order: 3
   metadata:
    httpsTargetOnly: false
    httpRuriOnly: false
    sbiRoutingEnabled: false
   filterName1:
     name: SbiRouting
     args:
       peerSetIdentifier: set0
       customPeerSelectorEnabled: false
       errorHandling:
        - errorCriteriaSet: scp_direct2_criteria_1
          actionSet: scp_direct2_action_1
          priority: 1
        - errorCriteriaSet: scp_direct2_criteria_0
          actionSet: scp_direct2_action_0
          priority: 2


- id: scp_direct2_criteria_1
    method:
      - GET
      - POST
      - PUT
      - DELETE
      - PATCH
    response:
      cause:
```

```
        ignoreCauseIfMissing: false
        path: ".cause"
        reason:
         - "cause-1"
         - "cause-2"
      statuses:
        - statusSeries: 4xx
          status:
            - 400


- id: scp_direct2_action_0
   action: reroute
   attempts:2
   blackList:
    enabled: false
    duration: 60000
```

`errorcriteria` can also be configured only with the status code. Following is the sample:

```
sbiRoutingErrorCriteriaSets:
 - id: scp_direct2_criteria_1
    method:
      - GET
      - POST
      - PUT
      - DELETE
      - PATCH
    response:
      statuses:
        - statusSeries: 4xx
          status:
            - 400
            - 404
        - statusSeries: 5xx
          status:
            - 500
            - 503
```

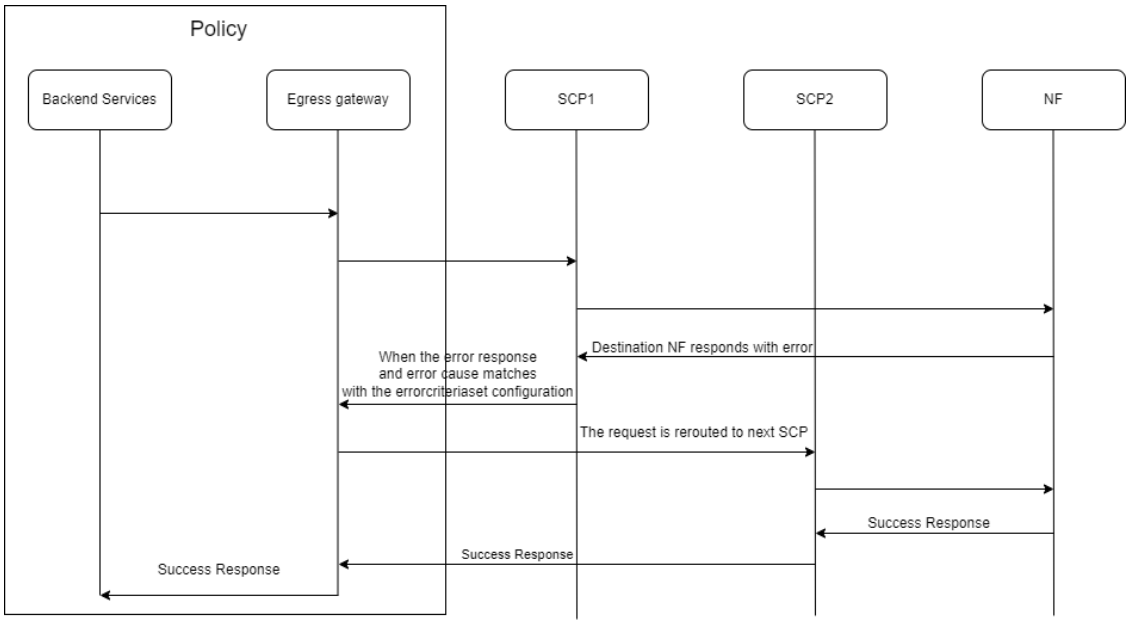
Following are the different use cases:

- When the HTTP response is received at the Egress Gateway only with the error code, then the error code is matched with the `errorcriteriaset` and rerouted to alternate SCP.
- When the HTTP response is received at the Egress Gateway with error code and error cause, then the error code and error cause is matched with the `errorcriteriaset` and rerouted to alternate SCP.
- When configured path is not found in the response and `ignoreCauseIfMissing` is true, then the response is received at Egress Gateway from SCP.
- When configured path is not found in the response and `ignoreCauseIfMissing` is false, then the response is sent back to Policy.
- When the cause is empty or null in the response body, `errorcriteria` is not considered as a match irrespective of `ignoreCauseIfMissing` is true or false and corresponding `erroractionset` is not executed.

For more information on configuration, see "SCP Configurations" in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.

The following diagram shows a scenario when the NF sends the error code and rerouting to alternate SCP is performed.

Figure 4-79 Session Retry at Egress Gateway



Policy sends the service request to the NF through SCP. When the NF sends the error code with cause to the Egress Gateway, the Egress Gateway compares the error code and error cause with `errorcriteriaset`. When it matches, it reroutes the request based on the erroractionset to the alternate SCP. If the maximum reattempts exhausts, the request is sent back to Policy.

Using Binding Headers, NFSets, and Server Headers

Policy supports session retry and alternate routing using Binding Headers, NFSets, and Server Headers.

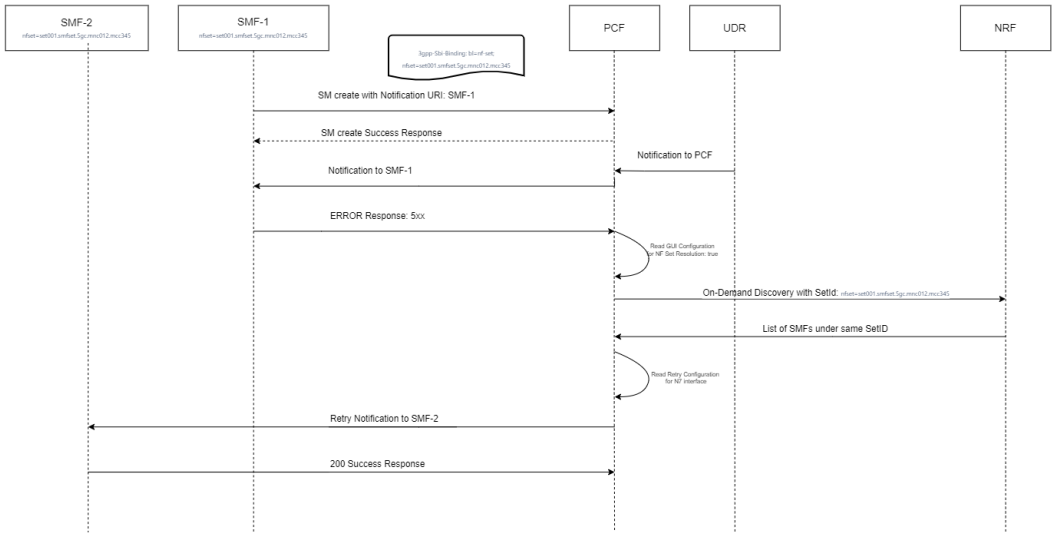
When indirect communication model is selected without delegated discovery, when the target NFs throw errors, SCP performs the task of reselecting target NFs to perform service operations. The reselection of alternate target NFs is based on various considerations such as binding headers, NFSets, and server headers sent by the HTTP client or consumer. In certain scenarios, reselection is based on error response and server headers as well. This feature also determines the action for Policy when no retry is required and all attempts are exhausted by SCP.

Session Retry for Notifications on N7 Interfaces

Policy supports the session retry functionality for failed notifications over N7 interface.

The following diagram depicts an example call flow for retry notification from UDR to SMF:

Figure 4-80 On demand NF Discovery is enabled (NF Set Resolution is set to NRF Discovery)



The call flow is explained as follows:

1. SMF 1 sends an SM Create request with the notification URI towards UDR. The request binding header contains the NFSetID.
2. After successful processing of the request, UDR sends a notification to SMF 1 through PCF.
3. On notification failure, SMF 1 sends a notification failure response with the error code to PCF.
4. PCF checks the Retry Profile configurations and if the value for the NF Set Resolution parameter under the Retry Profile configurations in Policy is set to **NRF Discovery**, then it performs an on demand discovery with SetID through NRF and receives the list of SMFs with the same SetID as SMF 1.
5. PCF retries sending the notification to SMF 2 based on the NFSetID list and receives *200 Success* response.

Managing Session Retry and Alternate Route Service

Enable

You can enable the Retry Profile and Alternate Routing functionality using the CNC Console or REST API for Policy.

- **Enable using CNC Console:** Enable the **Retry on Internal Send Failure** and **Enable Alternate Routing** parameters on the **Retry Profile** page. For more information about enabling the feature through CNC Console, see Retry Profiles.

  Retry Subscription Message enables the operator to choose Retry Logic for SUBSCRIBE/POST. For more information about retry subscription message, see PCF User Connector.
- **Enable using REST API:** Set the **enableRetry** and **enableAlternateRouting** parameter value to **true** in the **Retry Profile** configuration API. For more information about enabling the feature through REST API, see "Retry Profile" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

You can configure the Retry Profile and Alternate Routing functionality using the CNC Console or REST API for Policy.

- **Configure using CNC Console:** Perform the feature configurations on the **Retry Profile** page. For more information about configuring audit service, see Retry Profiles.
- **Configure using REST API:** Policy provides the following REST API for Session Retry and Alternating Routing configuration:
  API: {apiRoot}/oc-cnpolicy-configuration/v1/services/common/retryprofiles

  You can perform the POST, PUT, or GET operations to configure the feature. For more information about REST API configuration, see "Retry Profile" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

You can configure the parameter for alternate route service by updating the custom values file for Policy. For more information about configuring the parameter value, see "Alternate Route Service Configuration" in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide.*

Observe

Policy provides the Alternate Routing request and response metrics in the User Services. For more information, see User Service Metrics.

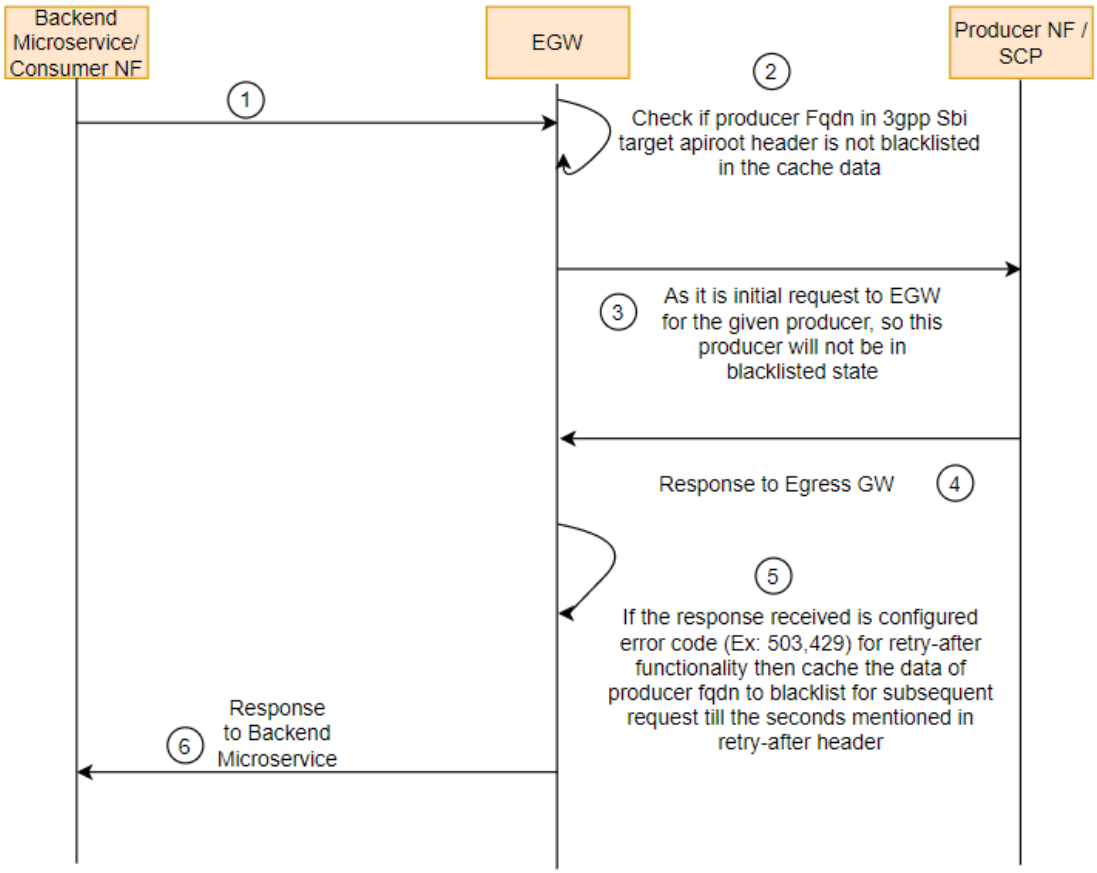
4.54 Support for Honor retry-after Header in Egress Gateway

This section describes the `retry-after` header functionality in Egress Gateway. This header is received as a response, from the Egress gateway. Based on the header information, the producer data is collected to block the producer FQDN for the number of seconds mentioned in the `retry-after` header.

Initial Call for Producer FQDN

The following call flow describes the initial call to Egress Gateway from the Producer FQDN:

Figure 4-81 Initial CallS for Producer FQDNs



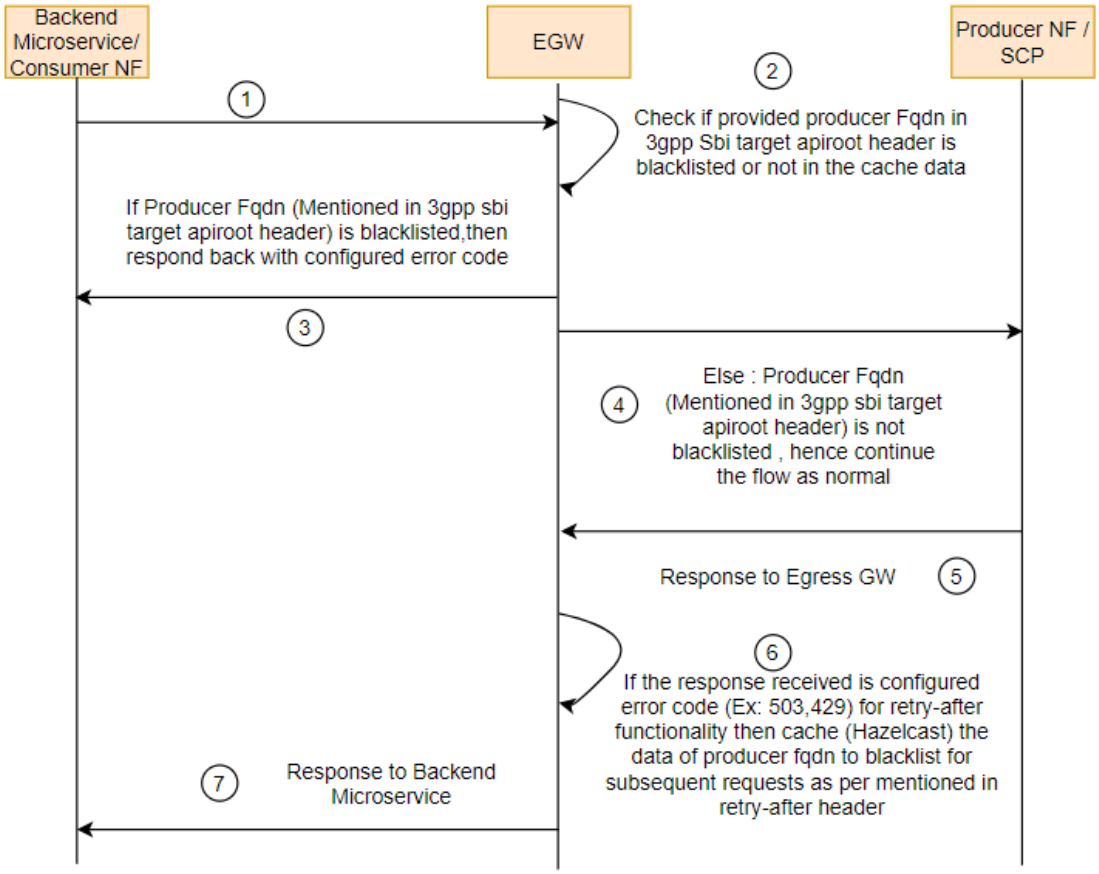
The initial call for producer FQDN is explained as follows:

1. The backend microservice or the consumer NF sends request to Egress Gateway.
2. The Egress Gateway verifies that the producer FQDN available in 3GPP SBI target apiroot header is not blocked.
3. Since, this is an initial request to Egress Gateway for the given producer, the producer FQDN is not blocked and the request is sent to the producer NF or SCP.
4. The producer NF or SCP sends a response to Egress Gateway.
5. If the response received by Egress Gateway is a configured error code, such as 503 or 429 for retry-after functionality, then the Egress Gateway cache the producer FQDN data to blocklist. It blocks the subsequent requests for the number of seconds mentioned in the `retry-after` header.
6. Egress Gateway sends a response to back end microservice or consumer NF that includes the `retry-header`.

Subsequent Call for Producer FQDN

The following call flow describes the subsequent call to Egress Gateway from the Producer FQDN:

Figure 4-82 Subsequent Calls for Producer FQDN



The subsequent call for producer FQDN is explained as follows:

1. The backend microservice or the consumer NF sends request to Egress Gateway.
2. The Egress Gateway verifies that the producer FQDN available in 3GPP SBI target apiroot header is not blocked.
   - If the producer FQDN available in 3gpp SBI target apiroot header is blocked, then the Egress Gateway responds back to the backend microservice or consumer NF with the configured returnErrorCode.
   - If the producer FQDN available in 3gpp SBI target apiroot header is not blocked, then the flow continues by sending request to producer NF / SCP.
3. The producer NF or SCP sends a response to Egress Gateway.
   - If the response received by Egress Gateway is a configured error code, such as 503 or 429 for retry-after functionality, then the Egress Gateway cache the producer FQDN data to blocklist. It blocks the subsequent requests for the number of seconds mentioned in the `retry-after` header.
   - If the response received by Egress Gateway does not have any configured error code, then no producer FQDN data is collected to be blocked.
4. Egress Gateway sends a response to backend microservice or consumer NF that includes the `retry-header`.

Managing Honor retry-after Header in Egress Gateway

Enable and Configure

To enable this feature in Policy, set the **egress-gateway.retryAfter.enabled** parameter to **true** under the `egress-gateway` configurations in the custom-values.yaml file for Policy.

Here is a sample configuration for this feature in custom-values.yaml file for Policy:

```
egress-gateway:
   retryAfter:
    enabled: true
    errorCodesToLook: 429,503
    #Provide error code with comma separated and no space
    returnErrorCode: 425
    #Error code expected to be sent to Backend NF when 3gpp-Sbi-Target-
    Apiroot header
    is provided with producer and request route/path which is blacklisted
    in egress
    gateway for a period mentioned in retry-after header
    blacklist-period: 0 #Seconds
    #This value is used when configured response code's are received from
    producer NF
    but retry-after header is absent. If retry-after value is configured
    with any
    value greater than 0 then it is considered in the absence of retryafter
    header from
    producer NF.
```

4.55 Stale Session Handling

Policy offers the signaling services such as, SM service, AM service, UE service, Binding Management service, Policy DS and, so on. These services are stateless in nature, thereby offloading session state to the centralized Oracle MySQL database (DB) for Policy.

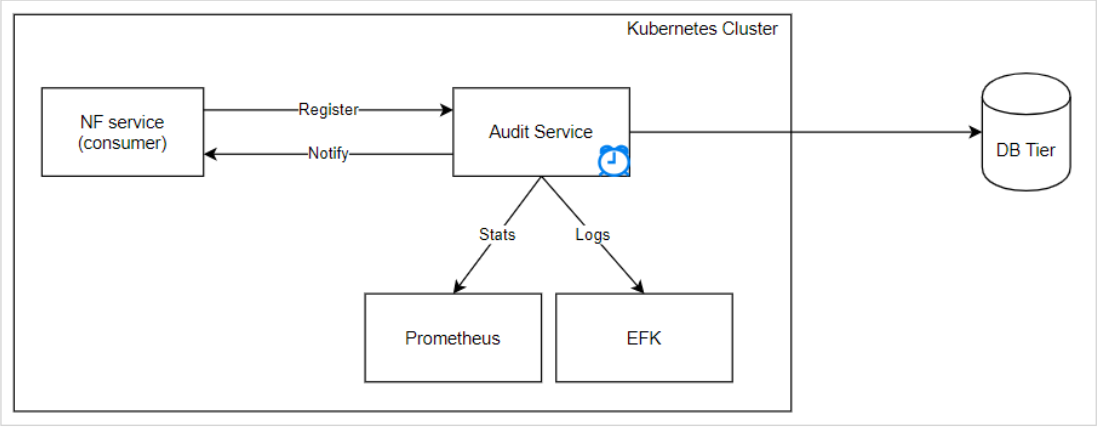
As the session processing microservices and the DB are different components that communicate over a network, there are chances for certain transaction failures during transit. For example, failures can occur in overload situations or as a result of code bugs. To manage such failed transactions, Policy provides a database audit mechanism that monitors the stale records and clean them up to not exhaust the database memory. The audit mechanism also notifies the microservice about the stale records so that the service can trigger signaling messages, if required. The stale session handling functionality ensures that the stale sessions are released by the consumer NFs. If applicable, the feature also releases the associated sessions from the same or other NFs. For example, deleting a stale SM association may require to delete the associated PA sessions.

> **Note**
>
> Policy supports the stale session handling for the following services:
>
> - SM service
> - AM service
> - PCRF Core
> - PDS
> - Binding service
> - UE Service
> - Usage Monitoring Service

Feature Design

The following diagram provides a high level design for stale session handling through session state Audit service:

Figure 4-83 High Level Design for Stale Session Handling using Audit Service



As shown in the above diagram:

1. The Policy service registers as a consumer with the Audit service and starts auditing the service database.
2. When a stale record is detected, the Audit service takes any of the following actions as requested by the Policy service during the registration:
   - Deletes the stale records from the database.
   - Sends a notification to the service about the stale records.
   - Deletes the stale records from the database and notify the service.
3. The Audit service implements a minimum wait time between consecutive audits and the consecutive notifications for the same record.

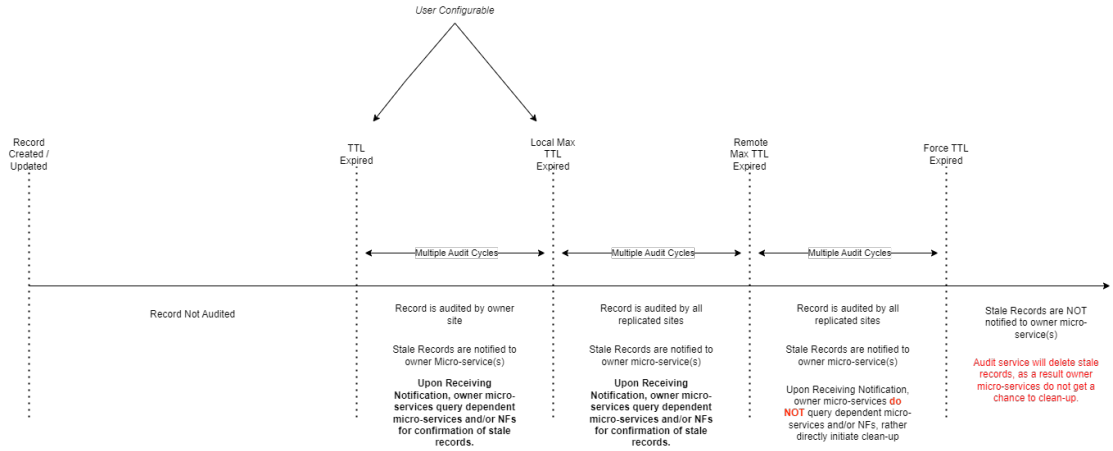
> **Note**
>
> - The Audit service stores the service registration details in the DB to be recovered after the pod restart or upgrade.
> - The audit service supports the local time zone.

Call Flow

Timeline View of Audit Service

The following figure shows a time line view of the Audit service for a record:

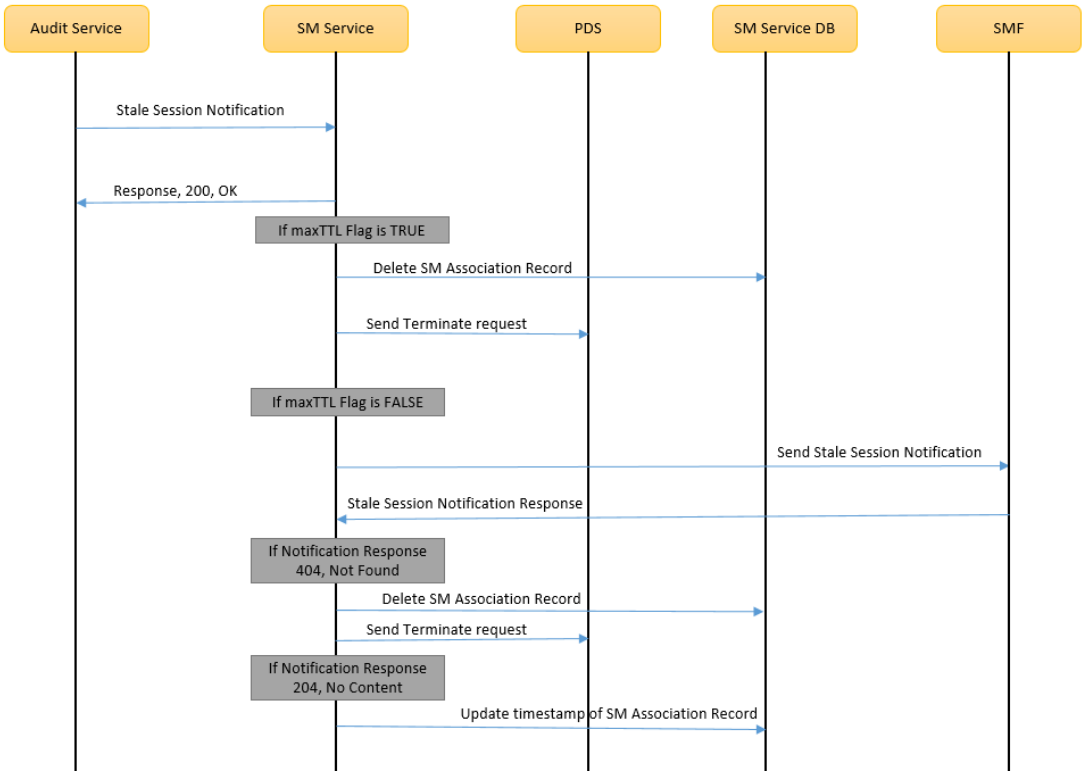
Figure 4-84 Timeline view of Audit Service



Stale Session Detection and Handling in SM Service

The following figure shows an example call flow of handling of stale session notification by the Policy SM service:

Figure 4-85 Example call flow for Stale Session Detection and Handling in SM Service



As shown in the call flow, after receiving a stale session notification from Audit service, SM service checks the `maxTtl` flag status and takes the following actions :

- If the `maxTtl` flag is **TRUE** then SM service triggers the TERMINATE action and deletes the record from the database.
- If the **maxTtl** flag is **FALSE** then SM service triggers another notification towards SMF to check if the particular SM association record exists with SMF:
  - If SMF responds with `204, No Content` status code then SM will **update** the timestamp of the SM Association record.
  - Else if SMF responds with `404, Not Found` status code then SM service triggers the TERMINATE action and deletes the record from the database..

Sending Terminate Notify to SMF

SM Service sends Terminate Notify to SMF when it detects and cleans up duplicate requests of SM PDU Session Create (Collision Detection) and stale SMPolicyAssociation (Max TTL reached for session).

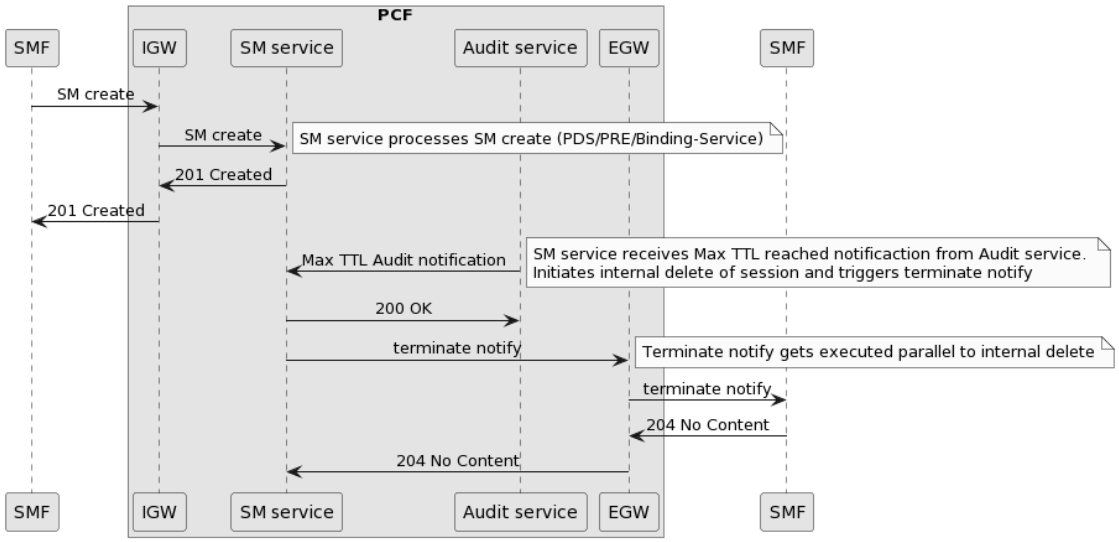
When Audit service sends a notification to SM service that Max TTL has reached and if the value of `AUDIT_SMPOLICY_ASSOCIATION.MAX_TTL.TERMINATE_NOTIFY.ENABLED` advanced settings key is set to `true`, the details are deleted from the SMPolicyAssociation database. Also, a Terminate Notify is sent to SMF. SMF performs the appropriate action to remove the stale SMPolicyAssociation.

Also, when a newer session is received, SM service checks if the value of `SYSTEM.COLLISION_DETECTION.TERMINATE_NOTIFY.ENABLED` advanced settings key is set to

`true`. It identifies the new session as duplicate and deletes the older session from SMPolicyAssociation database. Also, SM service sends a Terminate Notify to SMF.

For details on collision detection of duplicate Create session requests, see Detection and Handling of Late Arrival Requests.

Figure 4-86 Sample call flow for sending Terminate Notify when Max TTL notification is received from Audit service.



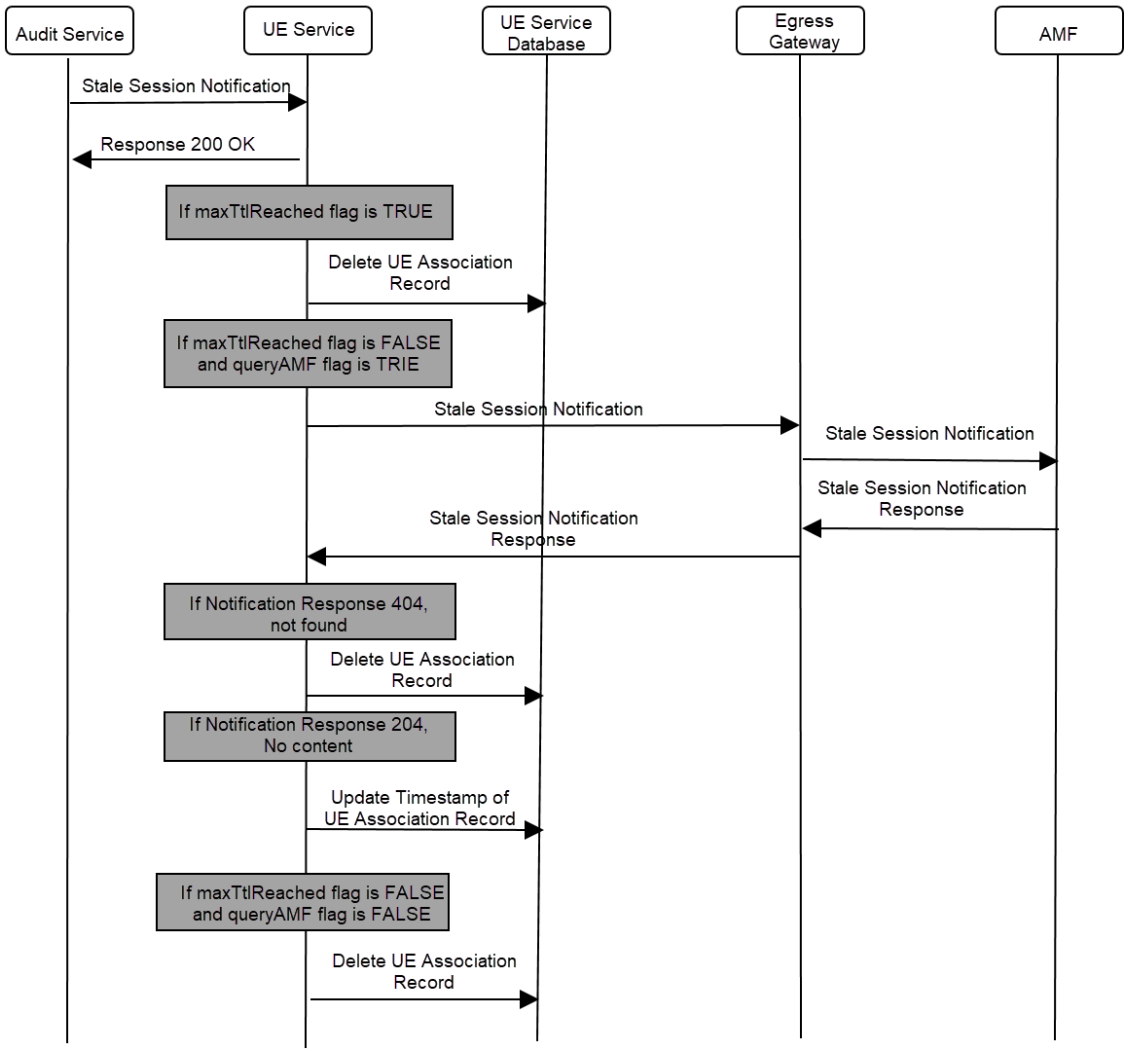
1. SM service receives a SM CREATE request from SMF through Ingress Gateway.
2. SM service sends a GET request to PDS to get the subscriber details from UDR, initiates a binding session request with Binding Service and then sends a request to PRE for policy evaluation. After the session is successfully created, SM services responds to SMF with 201 created message.
3. When the MaxTTL for the session is reached, the Audit service sends a MaxTTL Audit notification to SM service. SM service deletes the session details from its local SMPolicyAssociation database and then sends a Terminate Notify message to SMF through Egress Gateway.
4. After deleting the details of the older session, SMF responds to SM service with 204 No Content message.

Stale Session Detection and Handling in UE Service

A UE Policy Association on PCF is considered as stale when the association exists in PCF, but has no corresponding session on the AMF.

When a UE Policy Association is detected as stale, the UE service deletes the association from the database.

Figure 4-87 Example call flow for Stale Session Detection and Handling in UE Service



As shown in the call flow, after receiving a stale session notification from Audit service, UE service checks the status of `maxTtlReached` and `queryAMF` flags and takes the following actions:

- If the `maxTtlReached` flag is `TRUE`, the UE service triggers the TERMINATE action and deletes the record from the database.
- If the `maxTtlReached` flag is `FALSE` and `queryAMF` flag is `TRUE`, the UE Service sends the stale session notification to AMF via the Egress Gateway to check whether this particular UE Association record exists with AMF.

  The AMF replies with a stale session notification response to UE service.

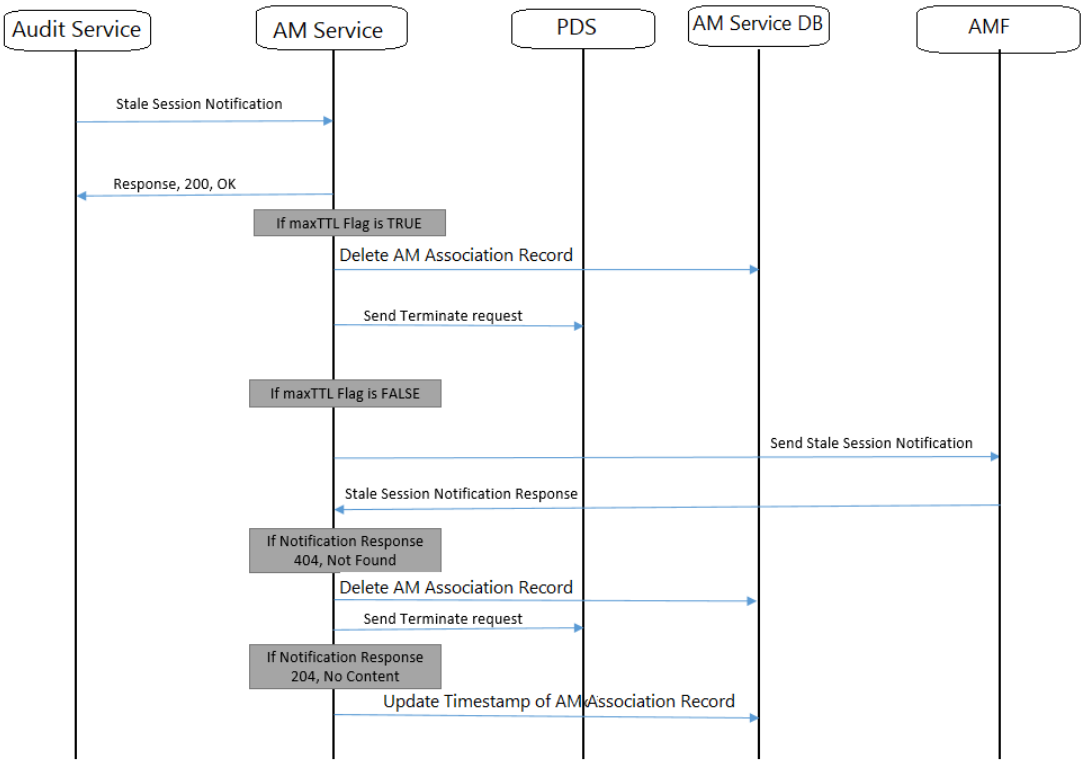
  - If the stale session notification response includes a `404 Not Found` status code, the UE Service triggers the TERMINATE action and deletes the record from the database.
  - If the stale session notification response includes a `204 No Content` status code, the UE Service updates the timestamp of the UE Association record in the database.

If either AMF does not respond, or is not reachable, no action is taken until `maxTtlReached` is reached.

- If the `maxTtlReached` flag is `FALSE` and `queryAMF` flag is `FALSE`, the UE Service triggers the TERMINATE action and deletes the record from the database.

Stale Session Detection and Handling in AM Service

Figure 4-88 Example call flow for Stale Session Detection and Handling in AM Service:



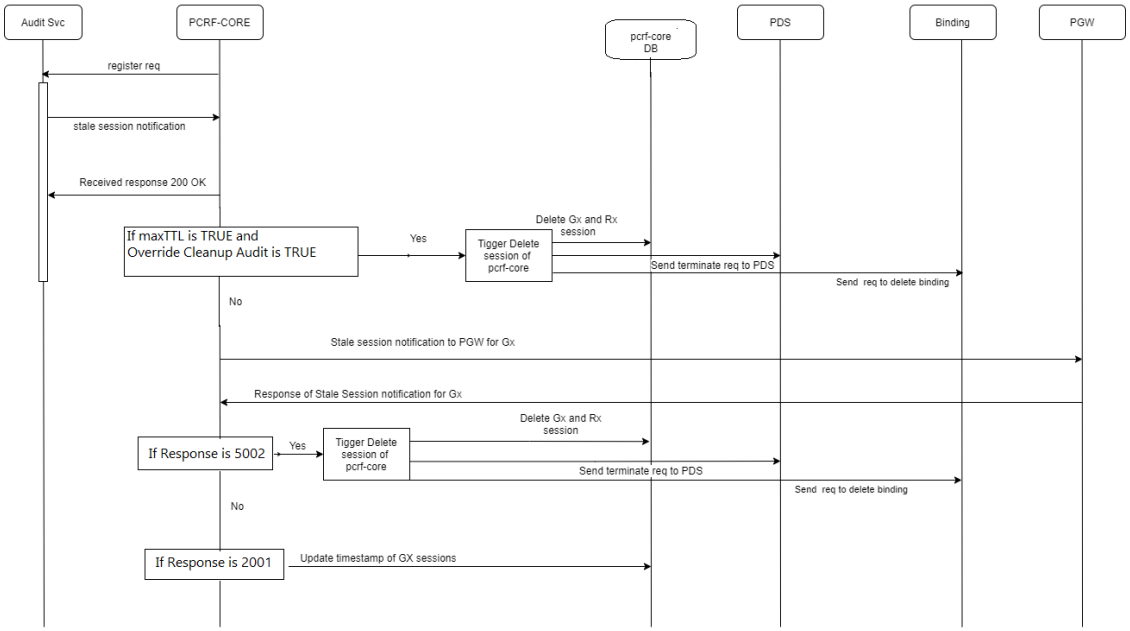
.

As shown in the call flow, after receiving a stale session notification from Audit service, AM service checks the maxTtl flag status and takes the following actions:

- If the **maxTtl** flag is **TRUE** then AM service will trigger it's **TERMINATE** leg.
- If the **maxTtl** flag is **FALSE** then AM Service will check the status of **queryAMF** flag.
- If **queryAMF** flag is set to true AM Service will trigger another notification towards AMF to check whether this particular AM Association record exists with AMF or not.
- If AMF responds with **204** No Content status code then AM will **update** the timestamp of the AM Association record.
- Else if AMF responds with **404** Not Found status code then AM Service will trigger its **TERMINATE** leg.
- Otherwise, if **queryAMF** flag is set to false then AM service will trigger it's **TERMINATE** leg.

Stale Session Detection and Handling in PCRF Core Service

Figure 4-89 Example call flow for Stale Session Detection and Handling in PCRF Core Service



Audit service audits the Gx sessions, Rx sessions and Sd sessions.

> **Note**
>
> PDS takes care of cleaning the Sy sessions.

Depending on the status of **maxTTL** flag, the Audit service send the notification to PCRF Core and initiates the stale session cleanup activity.

- For Rx session: If the **maxTTL** flag is **TRUE**, Rx sessions are cleaned.

  > **Note**
  >
  > The cleanup functionality is not implemented for Rx sessions, if the **maxTTL** flag is **false**.

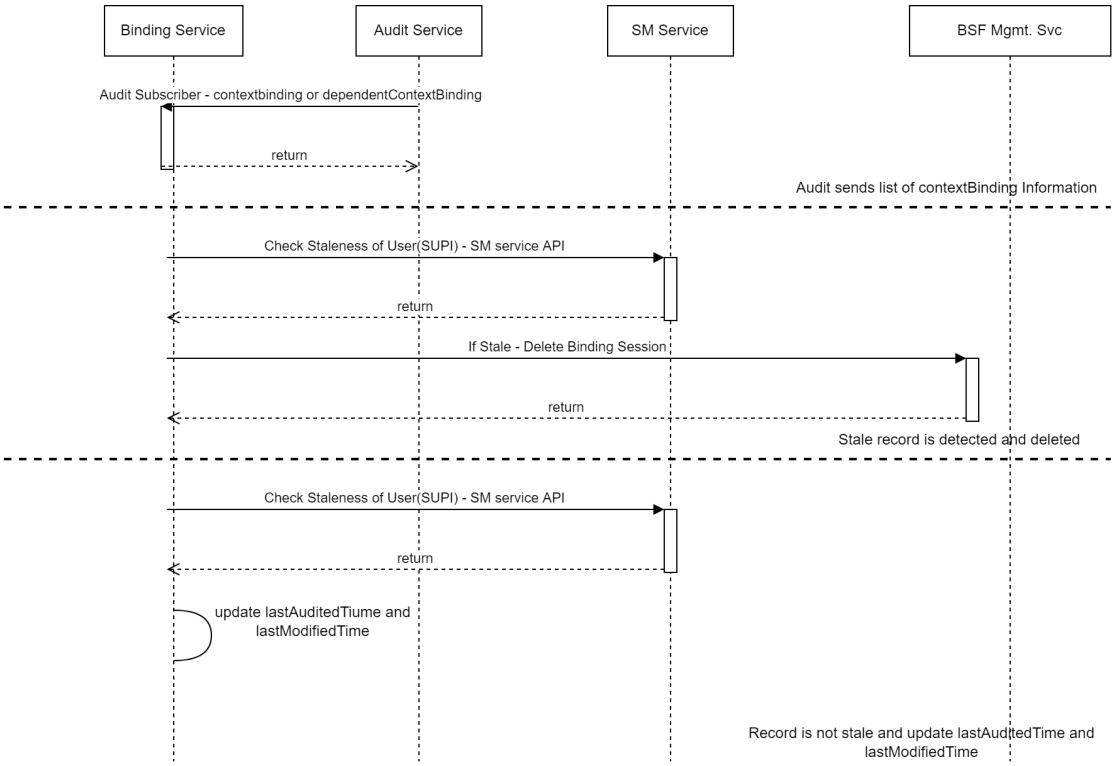
- For Gx session:
  - **DIAMETER.Cleanup.OverrideCleanupAudit** Advance settings key will also be checked along with **maxTTL** flag.
  - If both **maxTTL** and **DIAMETER.Cleanup.OverrideCleanupAudit** are **TRUE**, the Gx sessions are cleaned without querying the PGW. Dependent Rx sessions will also be deleted as part for cleanup.
  - If the **maxTTL** flag is **FALSE**,

    * pcrf-core service will trigger notification towards PGW to check whether this particular Diameter Session exists in the PGW or not.
    * If PGW responds with **2001 DIAMETER_SUCCESS** then pcrf-core will **update** the timestamp of the diameter session record.
    * Else, if PGW responds with **5002 DIAMETER_UNKNOWN_SESSION_ID** then pcrf-core Service will trigger its **TERMINATE** (CCR-T) leg.

- For Sd session: If the **maxTTL** flag is **TRUE** then Sd sessions are cleaned.

Stale Session Detection and Handling in Binding Service

Figure 4-90 Example call flow for Stale Session Detection and Handling in Binding Service



Audit Service uses the lastAuditedTime to check the staleness of the record.

1. For every **Minimum Audit Passes Interval** (frequency), Audit Service queries the database to get all the records that are older than the defined *TTl*.
2. If the records older than *TTl* have not yet reached *MaxTTl*, the Audit Service sends a audit notification to Binding Service with *maxTTl=false*. The Binding Service validates the staleness of the record with the contextOwner by sending request to core services.
    a. If the ContextOwner, has a valid session (Status code 200) - the binding service updates the *lastAuditedTimestamp*.
    b. Else, if the ContextOwner, does not have a valid session (Status code 404 from PCM-SM, 204 from PCRF-Core) - the Binding service deregisters the session as mentioned in the point 3.

3. If the records older than *TTl* and have reached *MaxTTl*, but not yet reached *maxTTLforceInterval*, the Audit Service sends a stale session notification to Binding Service.
   In case of ContextBinding, the Binding Service sends a deregister request to BSF, deletes the record in the Binding Service database and marks the dependent session (if any) as stale.

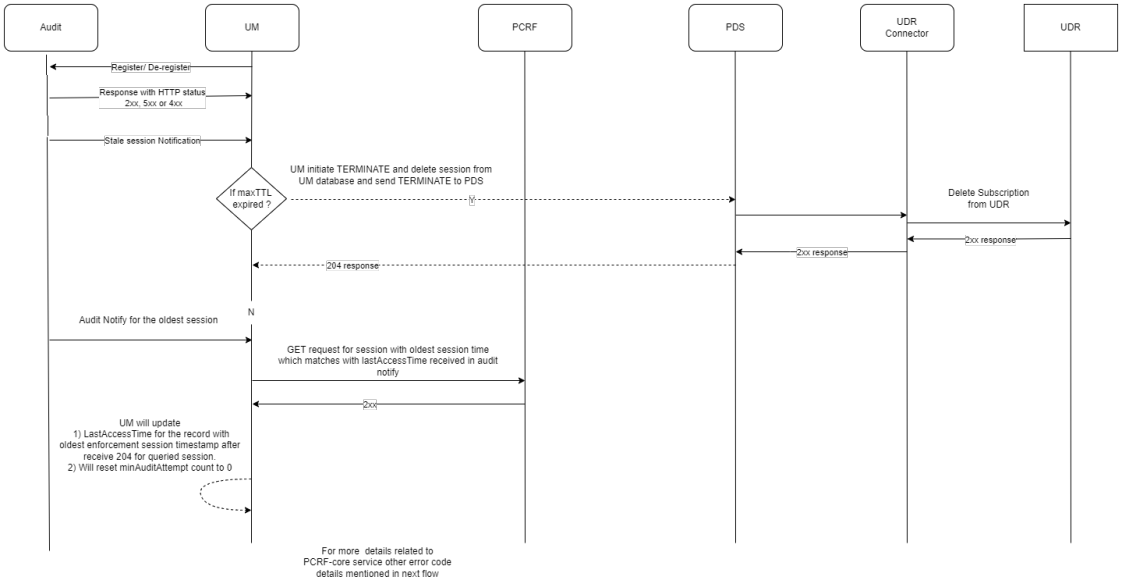
   In case of dependentcontextbinding, deletes the record from Binding Service database.
4. If the records older than the *TTl* and have reached *MaxTTl*, as well as *maxTTLforceInterval*, the Audit Service deletes the records in binding tables directly and no notification is sent.

> **Note**
>
> *maxTTLforceInterval* is configurable from deployment file of audit service. The Default value is of *maxTTLforceInterval* is 3 days (259200 seconds).

Stale Session Detection and Handling in Usage Monitoring Service

Usage Monitoring service identifies the stale sessions based on **lastAccessTime**.

Figure 4-91 Example call flow for Stale Session Detection and Handling in Usage Monitoring Service



1. The Usage Monitoring service sends register/de-register request to Audit service and takes necessary action on notifications received from Audit service.
2. Usage Monitoring service receives stale UM session notification from Audit service.
3. Usage Monitoring service checks the status of **maxTTL** flag.
4. If the value of **maxTTL** flag is `TRUE`, Usage Monitoring service initiates a UM session TERMINATE request for the oldest session to PDS to delete the session from UMContext database.

Usage Monitoring service deletes the record from its UmContext database only when it is the last session for that record.

5. Usage Monitoring service sends a GET request to PCRF Core to check if the Usage Monitoring session exists with PCRF Core service.
6. If PCRF Core service responds with **2xx** status code, the Usage Monitoring service updates the timestamp of the Usage Monitoring session for that subscriber, updates the **lastAccessTime** with the oldest session timestamp and resets the value of **Minimum Audit Attempt** field to 0.
7. If PCRF Core service responds with **404 Not Found** status code and the MaxTTL is expired, the Usage Montoring service triggers **TERMINATE** and also sends a DELETE request to PDS.
8. If PCRF Core service responds with **4xx/5xx** , except **404** , the Usage Monitoring service increaments the value of **Minimum Audit Attempt** count in the database.
9. Also, Usage Monitoring service sends a DELETE request to PDS, which in turn sends a DELETE request to UDR through UDR Connector to delete the subscription.

If **forceTTL** is triggered from Audit service, Audit service will delete the entire subscriber entry from Usage Monitoring (UMContext) database. Usage Monitoring service will not send the TERMINATE and unsubscribe request in that case.

For example, when there is a Usage Monitoring session for a subscriber that is qualified for forceTTL and the session is stale. In the interim, if Usage Monitoring service receives a new session with a different AVP for the same subscriber prior to the forceTTL being triggered, even then Audit service deletes the entire record for the subscriber.

Managing Stale Session Handling

Enable

You can enable the Stale Session Handling functionality using the CNC Console or REST API for Policy.

- **Enable using CNC Console:** To enable this feature, set the **Audit Enabled** parameter value to **true** on the **Audit Service** page. For more information about enabling the feature through CNC Console, see Audit Service.
- **Enable using REST API:** Set the **auditEnabled** parameter value to **true** in the **Audit Service** configuration API. For more information about enabling the feature through REST API, see "Audit Service" in *Oracle Communications Converged Policy REST API Specification Guide*.

Configure

You can configure the Stale Session Handling functionality using the CNC Console or REST API for Policy.

- **Configure using CNC Console:** To configure stale session handling feature for various services, perform the feature configurations under the **Audit** group on the respective service configurations page. For more information, see the following sections:
  - PCF Session Management
  - PCF Access and Mobility
  - Settings (for PCRF Core)
  - PDS Settings
  - PCF UE Policy Service

  - Configuring Usage Monitoring

  .

- **Configure using REST API:** Policy provides the following REST API for Stale Session Handling configuration:
  SM service: {apiRoot}/oc-cnpolicy-configuration/v1/services/pcfsm

  AM service: {apiRoot}/oc-cnpolicy-configuration/v1/services/pcfam

  PCRF Core: {apiRoot}/oc-cnpolicy-configuration/v1/services/pcrfcore/settings

  PDS: {apiRoot}/oc-cnpolicy-configuration/v1/services/pds/pdsSettings

  UE service: {apiRoot}/oc-cnpolicy-configuration/v1/services/pcfue

  You can perform the PUT operation to configure the feature. For more information about REST API configuration, see *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.
- Usage Monitoring service: {apiRoot}/oc-cnpolicy-configuration/v1/services/pcfum

Observe

Policy provides metrics and alerts specific to Audit Service. For information related to Audit Service metrics, see Audit Service Metrics.

Maintain

The Stale Session Handling functionality provides audit logging. At the end of each audit pass, an audit log is published on the Grafana dashboard with the following details of the pass:

- Database and Table audited
- Number of records found to be stale
- Number of records removed (for the DELETE action)
- Number of notifications sent (for the NOTIFY action)
- Time taken to complete the audit pass
- Any exceptions occurred

The following is a sample of the audit report for SM service:

```
Audit Report {
"database" : "pcf_smservice_161",
"table" : "SmPolicyAssociation",
"staleRecords" : 18869,
"recordsDeleted" : 0,
"timeToCompletePass" : 20,
"recordsEnqueuedForNotification" : 18869,
"exceptions" : [ ]
}
```

The following is a sample of the audit report for PDS:

```
Audit Report:
{
     "database" : "occnp_policyds",
     "table" :  "pdssubscriber",
     "staleRecords" : 1,
     "recordsDeleted" : 0,
```

```
    "timeToCompletePass" : 0,
    "recordsEnqueuedForNotification" : 1,
    "exceptions" : [ ]
 }
Audit Report:
 {
   "database " :  "occnp_policyds ",
   "table " :  "pdsprofile ",
   "staleRecords " : 1,
   "recordsDeleted " : 0,
   "timeToCompletePass " : 0,
   "recordsEnqueuedForNotification " : 1,
   "exceptions " : [ ]
}
```

The following is a sample of the audit report for UE Service:

4.56 Support for 3GPP NF Sets and Binding Headers

Policy supports the 3GPP NF Sets and binding headers in Model-B (Direct communication) and Model-C (Indirect communication). The support for 3GPP NF Sets and binding headers allows Policy to use the related headers and attributes for successful call processing.

> **Note**
>
> PCF can act both as an HTTP Server and HTTP Client.

NF Set: NF set is a group of interchangeable NF instances supporting similar services and network slices. In an NF set, the NF instances can be geographically distributed, but have access to the same context data. The NF instances can be deployed in such a pattern so that several instances are present within an NF set to provide distribution, redundancy, and scalability as a set. The NF instances of an NF set are equivalent and share the same MCC, MNC, NID (for SNPN), NF type, and NF Set ID.

Binding headers: The binding headers indicate the suitable target NF producer instance for NF service instance selection, reselection, and routing of subsequent requests associated with a specific NF producer resource or context. It allows the NF producer to indicate that the NF consumer, for a particular context, must be bound to an NF service instance, or NF set depending on local policies. Binding can also be used by the NF consumer to indicate the suitable NF consumer instances for notification target instance reselection and routing of subsequent notification requests, associated with a specific notification subscription.

PCF as a consumer sends binding headers to producer NFs such as UDR, CHF, BSF (customized), and AMF during explicit subscription. When **Send PCF Service Name in Binding Header** field under **NF Communication Profiles** page on CNC Console is enabled, the binding header includes `servname` along with other details such as `nfinst` and `nfset`.

When a notification request from any of the producer NFs such as UDR, CHF, BSF (customized), or AMF to PCF fails, the producer NFs use the details in the binding header to select alternate PCF. Using the service name included in the binding header, producer NFs can select alternate PCF with service level details.

Whenever there is an update in binding level and value, PCF includes the service name in the updated binding header sent to producers in Subscription Update or Notification Response.

Also, PCF sends binding header with `scope` as `callback`.

Support for N1N2 Message Transfer

PCF allows to configure sending of the binding header, routing binding header and the discovery header for N1N2 transfer using the below mentioned fields available under **AMF** group in **PCF UE Policy Service** page on CNC Console.

- Send SBI Binding Header For N1N2 Transfer Requests
- Send Routing Binding Header For N1N2 Transfer Requests
- Send Discovery Header For N1N2 Transfer Requests

Whenever there is a change in the binding information, UE Policy service sends the SBI binding in the transfer message. It can be configured whether N1N2Transfer must be sent with routing binding header or discovery header.

For N1N2 transfer requests UE Policy service sends the binding header as N1N2 transfer contains the implicit subscription for N1N2 failure notification.

- When NFSet and binding feature is enabled, PCF sends binding header in N1N2Transfer request at NFSet or NFInstance level, as configured by user.

  The user can configure to send SBI binding header for N1N2 Transfer requests by selecting one of the following values for **Send SBI Binding Header For N1N2 Transfer Requests** field:

  – **When binding information changes**: To send the binding header in the HTTP request only whenever there is a change in the binding information.

  – **Always**: To send the binding header in all the N1N2 Transfer requests.

  – **Never**: Not to send the binding header in any of the N1N2 Transfer requests.

  This configuration will override the values being set towards **NF Communication Profile** under Non-access stratum (NAS).

  > **Note**
  >
  > The NF Communication Profile for NAS is mandatory to be set in UE service configuration.

- The user can configure to send the routing binding header in N1N2Transfer requests by selectiing one of the following values for **Send Routing Binding Header For N1N2 Transfer Requests** field.

  – **Always**: To send the routing binding header in all the N1N2 Transfer requests.

  – **Never**: Not to send the routing binding header in any of the N1N2 Transfer requests.

- The binding header includes the discovery header when the N1N2 message transfer does not include the routing binding information. That is, UE Policy service adds 3gpp-Sbi-Discovery-target-nf-set-id when 3gpp-Sbi-Routing-Binding is not included in the binding information.

  The user can configure to send the discovery header in N1N2Transfer requests by selecting one of the following values for **Send Discovery Header For N1N2 Transfer Requests** field when the routing binding header is not present. This configuration is applicable to retransmissions as well.

  – **Always**: To send the discovery header in all the N1N2 Transfer requests when the routing binding header is not present.

– **Never**: Not to send the discovery header in any of the N1N2 Transfer requests.

For more details on how to configure the binding header, routing binding header, and discovery header for N1N2Transfer, see PCF UE Policy Service.

> **Note**
>
> It can be configured to include 3gpp-Sbi-Discovery-target-header in N1N2Transfer request explicitly and not in any other subsequent messages to AMF like Unsubscription of N1N2Subscription.

Table 4-27 Supported Headers

Header Name	Description
3gpp-Sbi-Binding	This header is used to communicate the binding information from an HTTP server for storage and subsequent use by an HTTP client. This header contains a comma-delimited list of Binding Indications from an HTTP server for storage and use of HTTP clients. The absence of this parameter in a Binding Indication in a service request is interpreted as "callback".
3gpp-Sbi-Callback	This header indicates if a HTTP request is a callback. For example, Notifications. This header is included in HTTP POST messages to request notifications or callbacks towards NF service consumers in indirect communication. This header is used by the SCP to differentiate the notification and callback requests from other service requests. For example, the authorization (access token is not used in notification or callback. This header contains the type of notification and the name of the notify service operation.
3gpp-Sbi-Discovery-*	The discovery parameter can be set using the CNC Console for Policy or the Policy REST API. The headers beginning with the prefix `3gpp-Sbi-Discovery-`, are used in indirect communication mode for discovery and selection of a suitable producer by the SCP. Such headers can be included in any SBI message and include information that allows an SCP to find a suitable producer, according to the delegated discovery parameters provided by the consumer. The name of each NF service discovery factors header are constructed by concatenating the string `3gpp-Sbi-Discovery-` with the name of the conveyed discovery parameter. For example, `3gpp-Sbi-Discovery-<discovery parameter>`. Policy supports the `target-nf-set-id` discovery parameter. Thus, the supported discovery header is `3gpp-Sbi-Discovery-target-nf-set-id`.
3gpp-Sbi-Routing-Binding	This header is used in a service request to send the binding information (3gpp-Sbi-Binding header) to direct the service request to an HTTP server, which has the targeted NF Service Resource context. This header enables alternate routing for subsequent requests at SCP. It contains a routing binding Indication to direct a service request to an HTTP server, which has the targeted NF service resource context.

Policy supports the NF Set and binding header functionality in SM, AM, UE, and PA function interfaces. You can configure NF communication profiles for the following PCF interfaces:

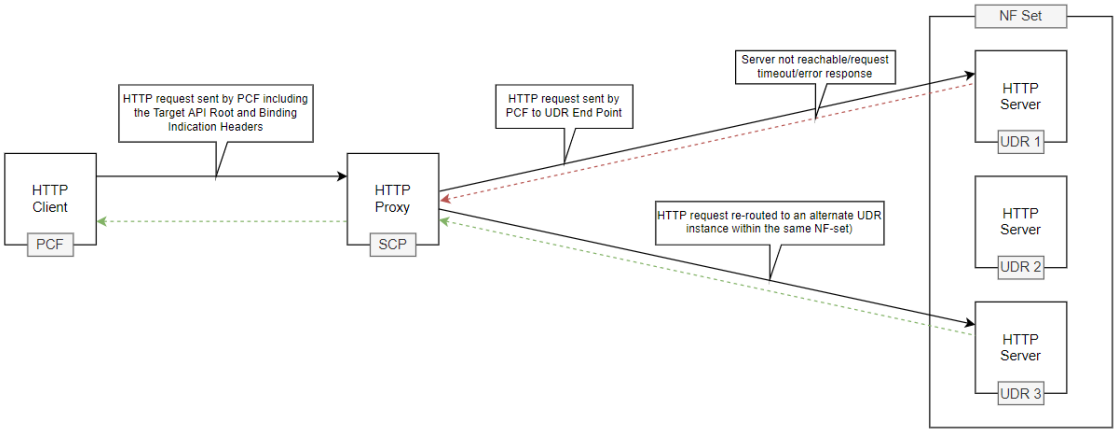
- N7/Npcf interface for notifications towards SMF
- N36/Nudr for requests and responses towards UDR
- N28/Nchf for requests and responses towards CHF
- Nbsf interface towards BSF
- N15/Namf interface for notifications towards AMF
- Npcf interface for notifications towards UE
- Npcf interface for notifications towards PA

You can configure the PCF to support binding header on the selected interface either as an HTTP client or server. If PCF is not configured to support the binding header for an interface, then it ignores the binding header and the information.

Example

The following diagram depicts an example of indirect communication between PCF and other NF, based on NF set and binding header:

Figure 4-92 Example of NF Set and Binding Header based Indirect Communication



The above architecture diagram describes a scenario of indirect communication between PCF and UDR. The PCF acts as an HTTP Client and UDR is the HTTP server. SCP acts as the HTTP proxy that redirects the requests coming from PCF.

In indirect communication, PCF sends the HTTP requests and the binding indication headers to SCP. The request contains the following details:

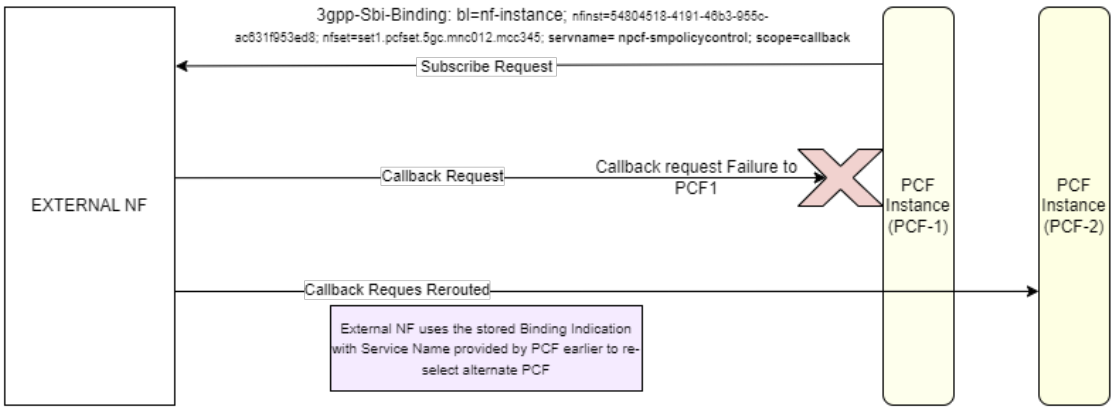
- 3gpp-Sbi-Target-Apiroot: The target API root that indicates the destination NF endpoint (Target UDR instance).
- 3gpp-Sbi-Discovery header: Contains information about the NF Set of the destination UDR instance. This header is used by SCP in case the first message does not reach the original UDR instance.
- 3gpp-Sbi-Routing-Binding: In case of an insertion request, PCF sends this header. It contains the routing binding indication to direct a subsequent service request to UDR.
- 3gpp-Sbi-Binding (PCF): Contains the NF Set information of PCF. This header is sent by PCF with request message and stored by UDR for subsequent communication.

- 3gpp-Sbi-Binding (UDR): Contains the NF Set information of UDR in the response mesage. This header is sent by UDR in response message and stored by PCF for subsequent communication.

Once the HTTP request is received at SCP, it is then redirected towards the NF instance indicated in the API root. In case of any failure, SCP redirects the request to an alternate destination belonging to the same NF set or a different NF set, as indicated in the *3gpp-Sbi-Binding* header.

The values can be set while configuring NF Communication Profile through CNC Console for Policy or REST APIs.

Support for Rerouting Notification from External NFs to Alternate PCF When the Notification Request Fails



PCF supports including the service name in the binding header sent to NFs such as UDR, CHF, AMF, and BSF (customized) when **Send PCF Service Name in Binding Header** parameter in the **NF Communication Profile** page is enabled.

The `servname` parameter in the binding header contains the name of the 3GPP defined PCF service (or aggregate service name applicable only towards CHF), which is invoking the Subscribe request or handling the notification. Invoking or Handler service name to be sent in the binding header is internally identified by PCF for UDR,AMF, and BSF interfaces and no external configuration is required except for the binding header sent towards CHF where the service name to be sent in the binding header is defined using CNC Console for Policy.

For example, the explicit Subscribe request sent towards nUDR on SM Create request for SM data contains `servname= npcf-smpolicycontrol` in the binding header as SM service is the handler service here.

Below is a sample Binding Header:

```
3gpp-Sbi-Binding: bl=nf-instance; nfinst=54804518-4191-46b3-955c-ac631f953ed8;
nfset=set1.pcfset.5gc.mnc012.mcc345; scope=callback;servname= npcf-
smpolicycontrol;
```

The service name to be forwarded in the binding header sent towards CHF is configured based on the value of **PCF Service name in Binding header** parameter under **CHF** group in **PCF User Connector** page under **Service Configurations**. The service name in the binding header sent to CHF contains:

- 3GPP defined PCF Service name: As PCF requests the policy counter information from CHF only once and can share with other services, PCF can contain the service name of the first service (for example, npcf-am-policy-control), which is requesting policy counter

information from CHF. The service level information such as scheme and port that the first service can be used to represent other services in the same PCF deployment as subsequent notifications from CHF will be consumed by all the services which is sharing the data.

Below is a sample binding header with 3GPP defined PCF service name sent towards CHF:

```
3gpp-Sbi-Binding: bl=nf-instance; nfinst=54804518-4191-46b3-955c-ac631f953ed8;
nfset=set1.pcfset.5gc.mnc012.mcc345; scope=callback;servname= npcf-am-policy-
control;
```

- Aggregate/custom service name representing a group of PCF services with aggregated service level information.

  As PCF (PCF1) requests the policy counter information from CHF only once and can share with other services, PCF can contain aggegrate service name approach for sending `servname` in the binding header. The aggegrate service level information is used to represent group of service with aggegrate service name.

  For example, npcf-custom-service can represent npcf-am-policy-control, npcf-smpolicycontrol, and npcf-ue-policy-control.

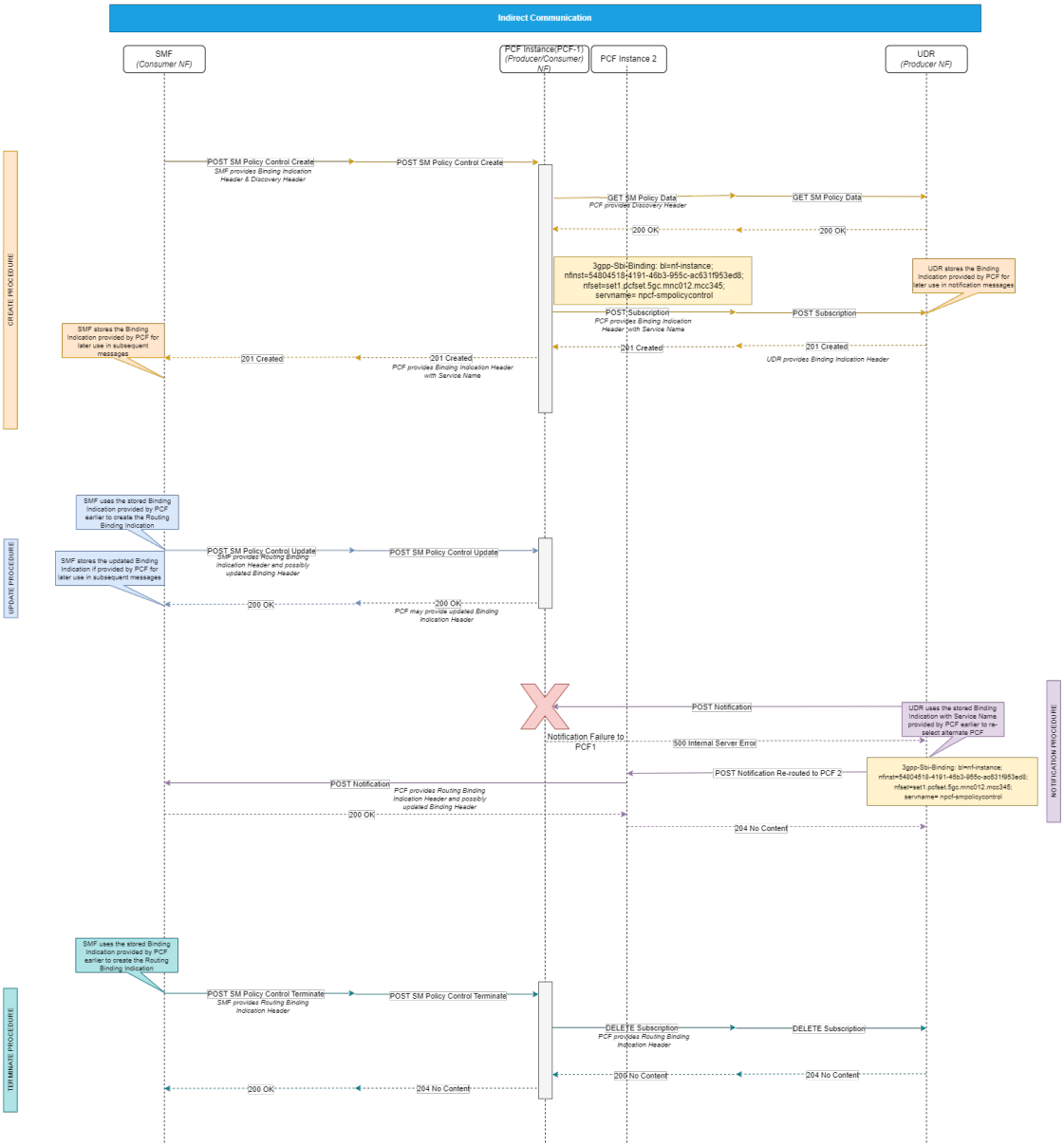
  Below is a sample binding header with aggregate/custom service name sent towards CHF:

  ```
  3gpp-Sbi-Binding: bl=nf-instance; nfinst=54804518-4191-46b3-955c-ac631f953ed8;
  nfset=set1.pcfset.5gc.mnc012.mcc345; scope=callback;servname= npcf-custom-
  service;
  ```

  > **Note**
  >
  > Before defining the aggregate/custom service name towards CHF, it is mandatory to register this custom service having aggregated service level information while doing PCF registration with NRF.

Call flow for rerouting a notification message from UDR to an alternate PCF using service name in Binding Header

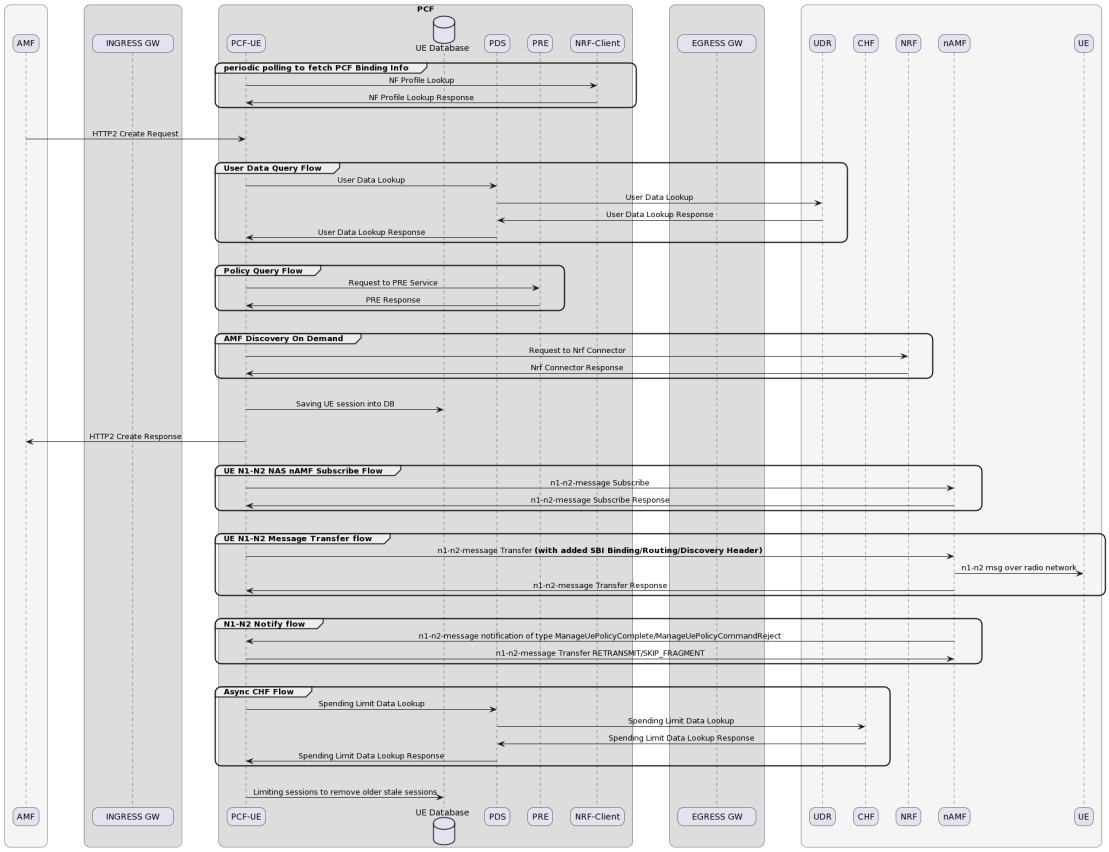
Here is a sample call flow that depicts rerouting a notification message from UDR to an alternate PCF using service name in the binding header when the NOTIFICATION request from UDR fails to reach PCF.



1. An instance of PCF, PCF1 receives the binding indication header and the discovery header in an SM Policy Control Create request from SMF (Consumer NF).
2. PCF1 queries UDR to get the corresponding SM Policy Data.
3. UDR sends a 200 ok message to PCF1.
4. Also, PCF1 sends the binding header and discovery header (3gpp-Sbi-Binding) in the Subscription request to UDR that includes `svcname` along with other parameters such as `nfinst` and `nfset`. UDR saves this binding information in its database, which will be used while sending any notification from UDR to PCF1.
5. When an update notification from UDR to PCF1 fails with a 500 internal server error, UDR identifies an alternate PCF2 using the service name provided by PCF in the previous binding header.
6. UDR reroutes the notification message to PCF2 and receives a response.

Support for Binding Header and Routing Binding Header in N1N2Transfer Request

Here is a sample UE Policy create call flow depicting Binding Header in N1N2Transfer request.



1. UE Policy service periodically polls the NRF through NRF client to fetch the updated NF profile.
2. Whenever UE Policy service receives a create request from AMF, it queries the UEPolicy Association database to fetch the UE Policy association details.
3. UE Policy service queries the UDR to fetch the user profile.
4. UE Policy service sends a request with the given details to PRE for Policy evaluation.
5. UE Policy service sends a request to NRF through NRF Client to discover the NF instances using amf-set-id and amf-region-id.
6. UE Policy service saves the updated UE Policy Session details in the UEPolicy Association database.
7. UE Policy service responds to AMF with a successful message.
8. UE Policy service sends a subscription request to nAMF for N1N2 message transfer and receives a response.
9. UE Policy service sends N1N2 message Transfer(with added SBI Binding/Routing/ Discovery Header) to nAMF for N1N2 message transfer, which in-turn forwards the details to the UE.

10. Whenever the UE Policy service receives an N1N2 notification of type ManageUePolicyComplete/ManageUePolicyCommandReject from nAMF, the UE Policy service tries to retransmit the message to nAMF.
11. Asynchronously, the UE Policy service contacts the PDF to fetach the Spending Limit data.
12. PDS contacts CHF through CHF Connector to subscribe for the policy counters.
13. After receiving the information from CHF, PDS sends a notification to UE Policy service with the SpendingLimit data.
14. UE Policy service processes this information and if needed it sends an update notify to AMF service and also saves the details in UEPolicy Association database.

Managing NF Sets and Binding Header Support

Enable and Configure

The NF Sets and Binding header support can be enabled for SM, AM, UE, and PA service interfaces. An NF communication profile must be configured and assigned to the respective interface to enable this feature.

You can enable and configure the NF Communication Profiles using the CNC Console or REST API for Policy.

- **Configure using CNC Console:** Perform the feature configurations under:
  - **NF Communication Profiles** page under **Common Data** for **Service Configurations**. For more details, see NF Communication Profiles
  - **AMF** group in **PCF UE Policy** page for **Service Configurations**. For more details, see PCF UE Policy Service.
  - Configure **NF Communication Profile** parameter for each of the services. For details, see:
    - PCF Session Management
    - PCF Access and Mobility
    - PCF Policy Authorization
    - Binding Service
    - PCF UE Policy Service
  - Configure the paramters under **UDR** and **CHF** groups under **PCF User Connector**. For more details, see PCF User Connector.
- **Configure using REST API:** Perform the NF Communication Profile configurations under the following REST APIs:
  - {apiRoot}/oc-cnpolicy-configuration/v1/services/common/nfcommprofiles
  - {apiRoot}/oc-cnpolicy-configuration/v1/services/pcfsm
  - {apiRoot}/oc-cnpolicy-configuration/v1/services/pcfam
  - {apiRoot}/oc-cnpolicy-configuration/v1/services/pcfpa
  - {apiRoot}/oc-cnpolicy-configuration/v1/services/pcfue
  - {apiRoot}/oc-cnpolicy-configuration/v1/services/binding
  - {apiRoot}/oc-cnpolicy-configuration/v1/services/pcfuser

  You can perform the POST, PUT, or GET operations to configure the feature. For more information about REST API configuration, see *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Observe

Policy uses the Ingress and Egress metrics for the NF bindings used by PCF. The following metrics contains the information about NF bindings used by PCF:

- For SM, PA, AM, UE, and Binding service:
  - ocpm_ingress_response_total
  - ocpm_egress_request_total
- For UDR Connector service: `ocpm_udr_tracking_request_total`
- For CHF Connector service: `ocpm_chf_tracking_request_total`

For more information, see the following sections:

- SM Service Metrics
- User Service Metrics

Maintain

The Policy logs includes the NF binding information received or sent by PCF. The logs include information about the following headers:

- `location`
- `3gpp-sbi-target-apiroot`
- `3gpp-sbi-binding`
- `3gpp-sbi-routing-binding`
- `3gpp-Sbi-Discovery-target-nf-set-id`
- `3gpp-sbi-callback`

4.57 Georedundancy Support

The Cloud Native Core architecture supports Geographical Redundant (Georedundant) Policy (PCF, PCRF, and converged Policy) deployments to ensure high availability and redundancy. It offers a two, three, or four-sites georedundancy to ensure service availability when one of the Policy sites is down.

The specifications for georedundancy feature are as follows:

- All the georedundant sites must have helm and REST based configurations except NF Instanced Id, Policy Endpoint, and port.
- The georedundant Policy sites must be reachable from NFs or Peers on all the sites.
- The same NFs or Peers must not communicate to other georedundant Policy sites at the same time for the same session.
- When Policy is deployed as georedundant PCF instances, then:
  - All the sites register with NRF independently and works in active state.
  - All Policy instances share the **Session State** data by using the replication service of DB tier. This enables service continuity during failure of any of the site.
  - The NFs in a given site can discover Policy instances through NRF. However, local configurations, such as DNS SRV or static configuration are required to determine the primary and secondary or alternate Policy configuration. When the primary instance is available, the NFs send service requests to the primary instance.

  - NRF always reflects current functional status of a given PCF or BSF instance. Thus, during failure of a given PCF or BSF instance, the value of `NfStatus` is updated to **SUSPENDED** by either NRF or PCF or BSF instance. Therefore, when NF detects failure of primary instance due to error response or status notification from NRF, the NF redirects its traffic to the secondary instance, while the primary instance remains unavailable and until it becomes available again.

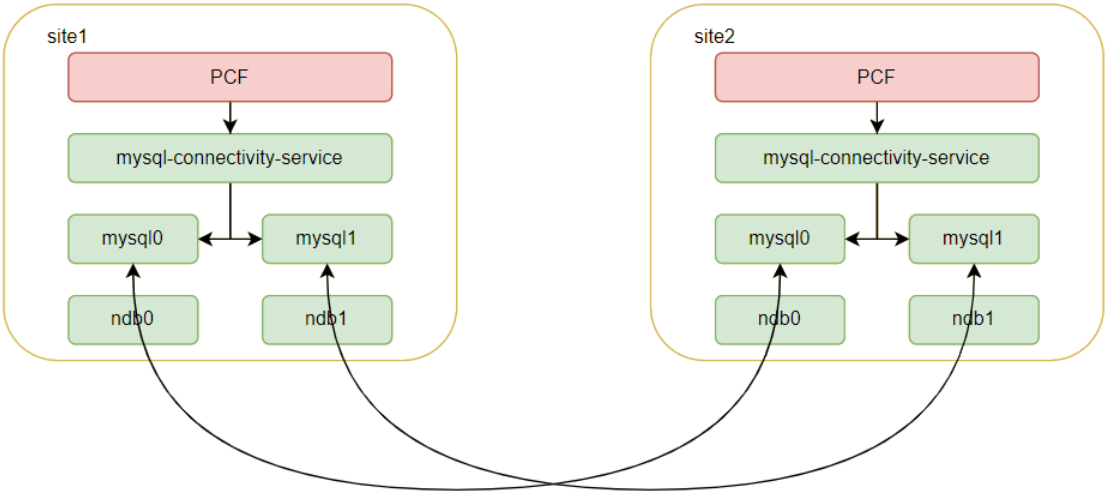
- When Policy is deployed as georedundant PCRF instances, then:
  - All the sites works in active state.
  - All instances share "Session State" data using DB tier's replication service to enable service continuity during site failure.
  - When peer detects failure of primary instance, the peer redirects its traffic to the secondary instance while the primary instance remains unavailable and until it becomes available again.

Policy supports the following types of georedundant deployment:

Two-Site Georedundancy Deployment

The following diagram depicts the topology for two-site georedundant PCF deployment:

Figure 4-93 Two-Site Georedundancy Deployment

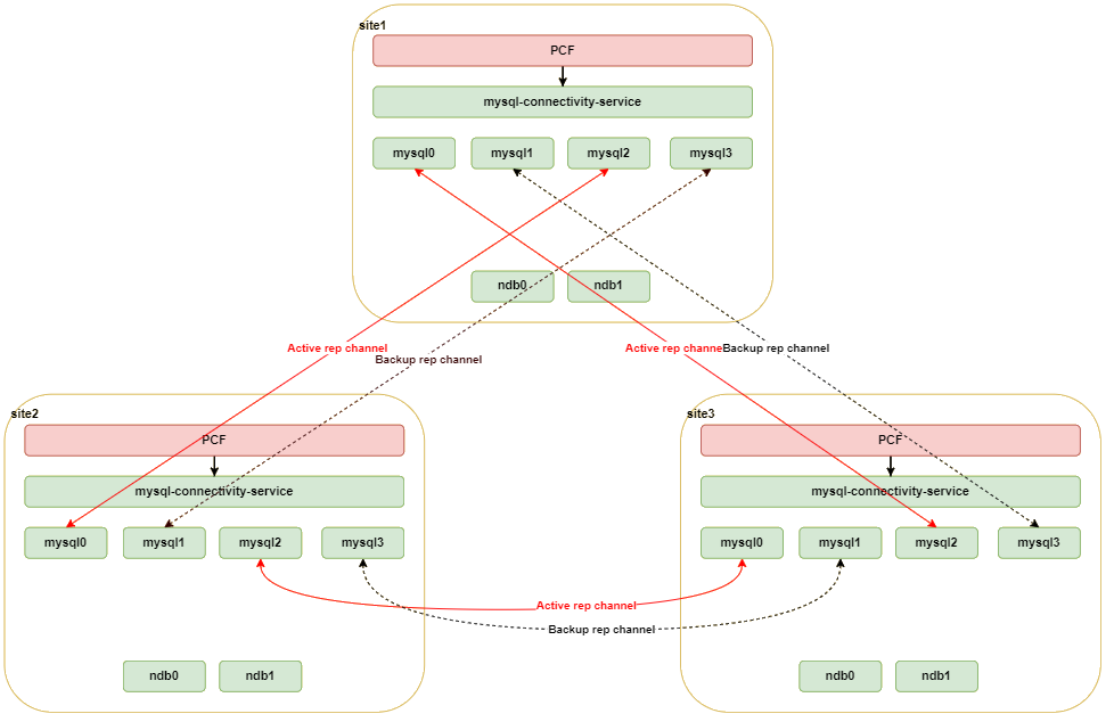


The two-site georedundancy is established, when the second site instance of the cnDBTier is created. The DB Tier provides bi-directional replication between both the sites. When the two-sites are correctly replicated, then any update done at one site is replicated to the other remote site in real-time. The changes include creating, changing, or deleting a record.

Three-Site Georedundancy Deployment

The following diagram depicts the topology for three-site georedundant PCF deployment:

Figure 4-94 Three-Site Georedundancy Deployment



In case of three-site georedundancy, bi-directional replication is established from each site to the two other sites. The database updates from each site are replicated to the other two-sites over the replication channel.

The advantages of three-site georedundancy is:

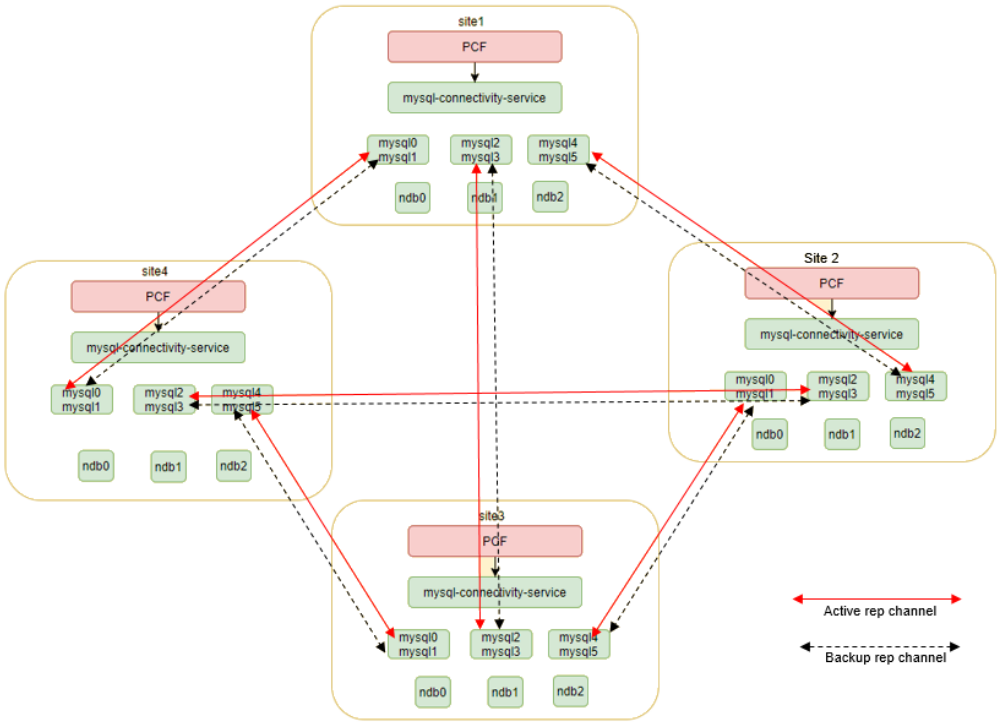
- In case of a single site failure, the remaining two-sites keep establishing the bi-directional replication.
- No action is required in case of a site failure.
- Requires 4 SQL pods and 2 replicated svc (db-rep-svc) at each site

When the three-sites are correctly replicated, then any update done at one site is replicated to the other two remote sites in real-time. The changes include creating, changing, or deleting a record.

Four-Site Georedundancy Deployment

The following diagram depicts the topology for four-site georedundant PCF deployment:

Figure 4-95 Four-Site Georedundancy Deployment



cnDBTier supports the four-site georedundancy deployment. In case of four-site georedundancy, each site participates in a 4-way replication. The database updates from each site are replicated to the other three-sites over the replication channels.

The advantages of four-site georedundancy is:

- In case of a single site failure, the remaining three-sites keep establishing the bi-directional replication.
- No action is required in case of a site failure.
- Requires 6 SQL pods and 3 db-rep-svc at each site.
- Each site uses two SQL nodes for active and standby replication channels for high availability of the replication channels.

When the four-sites are correctly replicated, then any update done at one site is replicated to the other three remote sites in real-time. The changes include creating, changing, or deleting a record.

Managing Georedundancy

Deploy

To deploy Policy in a georedundant environment:

1. Set up the replicated cnDBTier version 1.8.0.0.3 or above, on two or three-sites, as required. For information about installing cnDBTier, see "Installing cnDBTier" in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade and Fault Recovery Guide*.

2. Deploy Policy over the replicated (two or three) cnDBTier sites. For information about installing and deploying Policy, see *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide.*

Configure

To configure georedundancy:

You need to configure the georedundancy functionality while deploying the Policy instances on the replicated sites. The following parameters must be updated in the custom values file for Policy:

Table 4-28 Georedundancy Parameters

Parameter	Description
global.envMysqlHost	The database instance for each site. PCF communicates to the database at the same site only.
global.nfInstanceId	The ID for the site
config-server.envMysqlDatabase	The database for the config server. The two-sites should use different database names for config server
cm-service.envCommonConfigMysqlDatabase	The common configuration database. It should be different on the sites
nrf-client.configmapApplicationConfig.profile	Configuration data for nrf client. The `appProfile` and the `nfInstanceId` parameters must be aligned with `global.nfInstanceId`
nrf-client-nfdiscovery.dbConfig.dbName	The common configuration database. It should be different on the sites
nrf-client-nfmanagement.dbConfig.dbName	The common configuration database. It should be different on the sites
appinfo.dbConfig.dbName	The common configuration database. It should be different on the sites
perf-info.dbConfig.dbName	The common configuration database. It should be different on the sites
policyds.envMysqlDatabaseConfigServer	The database for the config server. The two-sites should use different database names for config server
ingress-gateway.dbConfig.dbName	The common configuration database for ingress gateway. It should be different on the sites
egress-gateway.dbConfig.dbName	The common configuration database for egress gateway. It should be different on the sites
alternate-route.dbConfig.dbName	The common configuration database for alternate route. It should be different on the sites

For more information about configuring the parameter value, see "Alternate Route Service Configuration" in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide.*

Observe

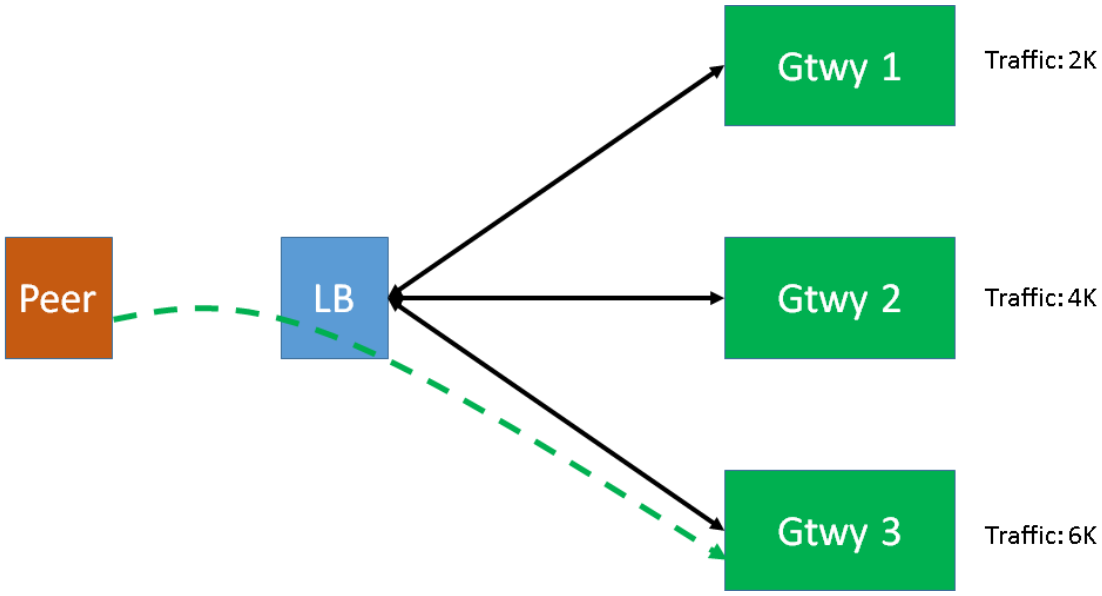
cnDBTier generates critical alerts in case of application or database failure. For more information, see *Oracle Communications Cloud Native Core, cnDBTier User's Guide*.

Maintain

Policy allows you to monitor the georedundancy deployment through cnDBTier alerts. Access the Prometheus GUI to check for new App alerts.

4.58 Diameter Pod Congestion Control

The Diameter Gateway is a diameter proxy agent for Policy. Being a front-end microservice for diameter traffic for both Ingress Gateway and Egress Gateway, the Diameter Gateway can get congested due to higher traffic, higher CPU usage, and higher memory utilization. Thus, it is imperative to have suitable congestion control features in place for Diameter Gateway pods to avoid adverse impact on latency and performance. Another reason for the need of congestion control mechanism for Diameter Gateway is the nature of diameter connections. These connections are long lived and are distributed by external LoadBalancer. As shown in the following image, when LoadBalancer routes an incoming request from network to Diameter Gateway pod, it does not take into account health or load of the pod. As a result, uneven distribution of traffic may happen and gateway pods experience congestion.

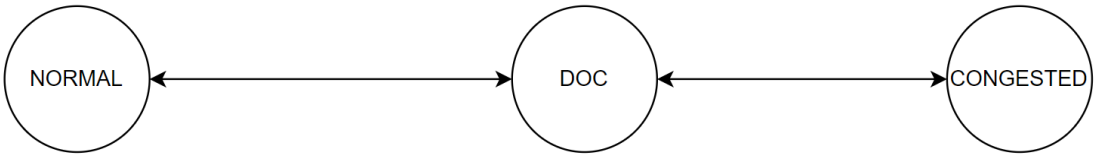


Determining Pod Congestion State

At any given point of time, a pod can be in any one of the following states:

1. Normal
2. DOC (Danger of Congestion)
3. Congested

Figure 4-96 Congestion states



To decide the state of a pod, the following points are taken into consideration:

1. Calculate the congestion state for the following resources:
   a. **Queue**: Compare the count of pending messages (in the queue) against the configured pending messages threshold for each congestion state (DOC, CONGESTED).
   b. **CPU** - CPU usage for congestion state is calculated by comparing the CPU usage of the container (monitored using cgroup parameter - `cpuacct.usage` that provides current cpu usage in nano seconds) with the configured threshold. The following formula is used to calculate CPU usage:

$$\frac{\frac{CurrentCpuUsage - LastCpuUsage}{CurrentTime - LastSampleTime} * 100}{CPUs}$$

   c. **Memory** - Memory usage for congestion state is calculated by comparing the memory usage of the container with the configured threshold. The following formula is used to calculate memory usage:

$$\frac{memoryUsage}{memoryLimit} * 100$$

   where memory limit is monitored using `memory.limit_in_bytes` cgroup parameter and current memory usage is monitored using `memory.usage_in_bytes` cgroup parameter.
2. Based on the congestion state of resources, the congestion state for the pod is set to the maximum of congested states. The following table describes how the state of the pod is evaluated for various scenarios:

Table 4-29 Published Pod Congestion State

Queue	CPU	Memory	Pod
CONGESTED	NORMAL	DOC	CONGESTED
NORMAL	DOC	NORMAL	DOC
DOC	DOC	Normal	DOC

3. The current published congestion state of the pod is changed to calculated congestion state only when the calculated state remains same for the configured number of continuous sample counts (100 ms by default). By doing so, events like short bust of traffic triggering a change in the congestion state and load shedding can be avoided. The only exceptions to this rule is when current state is NORMAL and calculated state is CONGESTED, and current state is CONGESTED and calculated state is NORMAL.

Triggering Congestion Control

Every time a message is fetched from the ring buffer for processing, the system checks the current congestion state of the pod. If the current state is either DOC or Congested, the

congestion control mechanism is invoked. After verifying that the message type is a request, a priority is assigned to it. If the assigned priority is less than or equal to discard priority, the message is rejected.

> **Note**
>
> Congestion control does not apply to response messages as they are always accepted.

Figure 4-97 Process flow for triggering congestion control

The priority value for a request can be from 0 to 15 where 0 is the highest priority and 15 is the lowest priority.

For priority based load shedding to happen, load rule configured for current congestion state is taken into consideration. If there is no rule configured for a congestion state, the requested is accepted by default. While defining load rules, the result to reject the request can also be customized by the user. However, to reject requests from backend peers, the result code is always DIAMETER_TOO_BUSY.

Call Flow

To best understand how the diameter congestion control feature works, consider a scenario where the sample message priority profiles rules are configured as described in the following table:

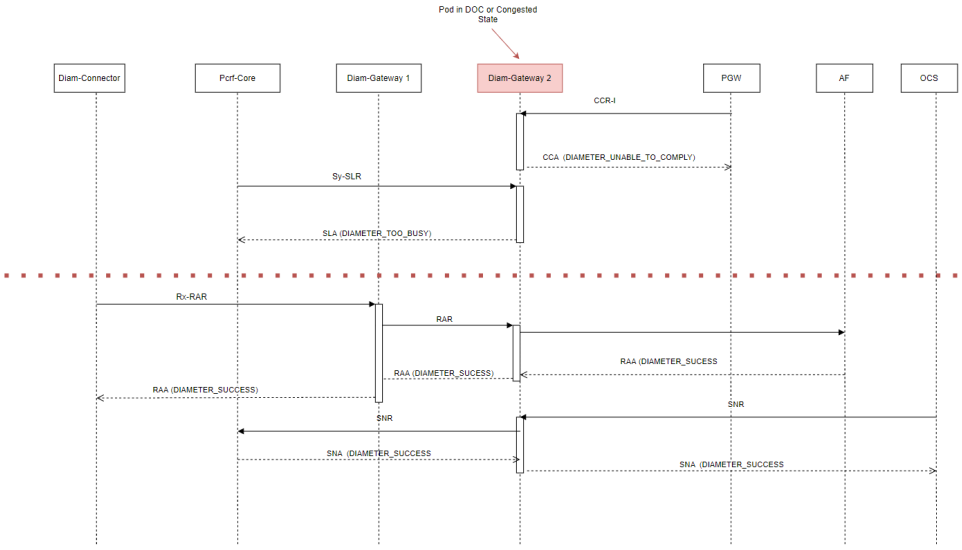
Table 4-30 Message Priority Profiles

Message	Priority
Default	6
Rx RAR	7
Gx CCR-I	10
Sy SLR	11

> **Note**
>
> The default message profile is only for the purpose of understanding the feature. No default message priority profiles are configured for Policy.

For this call flow, discard priority is set as 10 for DOC and Congested state. The result code for rejecting request messages is set as DIAMETER_UNABLE_TO_COMPLY (applicable only for external peers).

Figure 4-98 Call flow diagram for Diameter Congestion Control



- When Diameter Gateway 2, currently in DOC or Congested state, receives CCR-I request, the priority is compared against discard priority. As both message priority and discard priority values are same, the message is rejected with result code – DIAMETER_UNABLE_TO_COMPLY.
- When Diameter Gateway 2, currently in DOC or Congested state, receives Sy-SLR request, the priority is compared against discard priority. As message priority is low in priority than discard priority, the message is rejected with result code – DIAMETER_TOO_BUSY.
- When Diameter Gateway 2, currently in DOC or Congested state, receives Rx RAR request, the priority is compared against discard priority. As message priority is higher than discard priority, the message is accepted with result code – DIAMETER_SUCCESS.
- When Diameter Gateway 2, currently in DOC or Congested state, receives SNR request, the priority is compared against discard priority. Since no priority rule is configured for this message request, the message is accepted with result code – DIAMETER_SUCCESS.

Enable

The congestion control for diameter gateway pods is a functionality supported by Policy configurations. You do not need to enable or disable this feature.

Configure

You can customize the configurations related to this feature using the CNC Console or REST APIs for Policy.

- **Configure using CNC Console:** Perform the feature configurations on the Load Shedding Profiles and Message Priority Profiles page. For more information about the configurations, see Diameter Configurations.
- **Configure using REST API:** Policy provides the following REST API for configurating diameter gateway pod congestion control feature:
  Load Shedding Profiles: {apiRoot}/oc-cnpolicy-configuration/v1/diameter/loadsheddingprofiles

Message Priority Profiles: {apiRoot}/oc-cnpolicy-configuration/v1/diameter/messageprioritypro files

Congestion Threshold: {apiRoot}/oc-cnpolicy-configuration/v1/threshold/{serviceType}

You can perform the PUT, or GET operations to configure the feature. For more information about REST API configuration, see *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Observe

Metrics

Policy uses the pod congestion metrics for this feature. For more information, see the Pod Congestion Metrics section. Alerts are raised when the following metrics are pegged:

- pod_congestion_state
- pod_resource_congestion_state

> **Note**
>
> Prometheus automatically injects name of the pod with label name "kubernetes_pod_name" to metric. This information is further used for alerting purposes.

Alerts

Policy uses the following congestion control alerts for this feature:

- PodDoC
- PodCongested
- PodPendingRequestDoC
- PodPendingRequestCongested
- PodCPUDoC
- PodCPUCongested
- PodMemoryDoC
- PodMemoryCongested

For more information, see the PCF Alerts section.

Maintain

Error logs are generated when the system is congested and the actions taken to bring the system back to normal. Warning logs are generated to indicate the congestion level. However, error logs are not generated when messages are rejected to avoid additional resource usage to write error logs.

The following is a sample error log:

```
{
  "instant": {
    "epochSecond": 1629306402,
    "nanoOfSecond": 438895435
  },
  "thread": "Thread-10",
  "level": "INFO",
```

```
  "loggerName": "ocpm.cne.common.configclient.ConfigurationAgent",
  "message": "Configuration changed, class ConfigurationItem {\n    key:
common.congestionthreshold.diam-gateway\n    value:
{\"stateCalculationInterval\":10000,\"stateChangeSampleCount\":3,\"thresholds\
":[{\"state\":\"DANGER_OF_CONGESTION\",\"resourceUsageLimit\":
{\"cpu\":40,\"memory\":40,\"queue\":20}},
{\"state\":\"CONGESTED\",\"resourceUsageLimit\":
{\"cpu\":80,\"memory\":80,\"queue\":50}}]}\n    version: 8\n    topicInfo:
class TopicInfo {\n        id: 6\n        name:
common.congestionthreshold.diam-gateway\n        description: Created by
config server.\n        modifydate: Wed Aug 18 17:06:42 UTC 2021\n
version: 10\n    }\n    labels: null\n}",
  "endOfBatch": false,
  "loggerFqcn": "org.apache.logging.slf4j.Log4jLogger",
  "threadId": 77,
  "threadPriority": 5,
  "messageTimestamp": "2021-08-18T17:06:42.438+0000"
}
```


4.59 Overload Control

Overload means when 100% of the planned capacity is exhausted. It can be due to uneven distribution of traffic towards a given policy service instance, network fluctuations leading to traffic bursts or unexpected high traffic volume at any given point of time.

During overload conditions, the service response times may grow to unacceptable levels, and exhaustion of resources can result in downtime or services exhibiting unexpected behavior. Overload management is a critical requirement for any telecom node, server, and service to protect against downtime and ensure serviceability during extreme overload conditions. Thus, the goal of overload management is to prevent service performance from degrading in an uncontrolled manner under heavy load. When policy service starts approaching its saturation or planned limit, response times typically grow high and throughput can degrade substantially. Under such conditions, it is desirable to shed load based on the operator's configuration, instead of causing all messages and signaling flows to experience unacceptable response times, failures, or downtime.

Policy allows to configure a percentage of messages to be rejected. That is, messages are discarded based on configured percentage. This enables system's overload and congestion control to manage gauge system's load with better accuracy. Also, it allows the user to provide less rejections instead of providing 100% rejections.

Percentage of message rejections for each load level is configurable. Also, the rejection percentage for each message priority can be configured.

For example, if the discard value for CCR-I messages is 50% , when system is under load, only alternate CCR-I requests are processed rejected the rest. That is, 1st CCR-I is rejected and 2nd is accepted.

> **Note**
>
> All CCR-Ts are accepted.

Enable

To enable the overload control functionality, set value for the following parameter to `true` in the `custom-values.yaml` file for Policy:

- **perf-info.overloadManager.enabled**

Then, configure the values for the following parameters in the `custom-values.yaml` file:

```
perf-info:
  envMysqlDatabase: ''
  overloadManager:
    enabled: false
    ingressGatewaySvcName: occnp-ingress-gateway
    ingressGatewayPort: *svcIngressGatewayHttp
    # nfType is used to query configuration from common cfg server
    nfType: PCF
    # diam Gateway overload management feature configurations
    diamGWPort: *svcDiamGatewayHttp
```

For more information about setting the parameter values, see *Overload Manager Configurations* in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*.

Configure

You configure the overload control feature either using CNC Console, or through REST API.

- **Configure using REST API:** Policy provides `overloadLevelThreshold` and `overloadLevelThresholdProfiles` API end points to configure overload control feature. You can perform the POST, PUT, or GET operations to configure the feature. For more information about REST API configuration, see *Overload Level Threshold Overload Level Threshold ProfileS* sections in *Oracle Communications Cloud Native Core, Converged Policy REST API Specification Guide*.
- **Configure using CNC Console**: Using CNC Console, you can configure the threshold values based on profiles. For more information, see Overload Control Threshold.

Overload Threshold Values

When Overload control feature is enabled, the load on each service is monitored using the following counters:

- CPU: Indicates the maximum CPU usage in terms of percentage.
- Pending message count: Indicates the number of requests from the gateway to the Policy service for which the response is not yet received.
- Failure count: Failed requrest count for each service. Indicates the number of error responses that the Policy service receives in error series such as 4XX or 5XX. Different routes mapped to the Policy service can have different set of configured error codes and exception list for tracking the failure counts.
  The error codes set must be configured in the routes configuration. The below table lists the route IDs for each of the services for which the error codes set must be configured.

Table 4-31 Supported Route IDs

Service	Routes
SM Service	– sm_create_session_route – sm_delete_session_route – sm_update_session_route – pa_create_session_route – pa_delete_session_route – pa_update_session_route – pa_events_subscription_route
AM Service	– am_create_session_route – am_delete_session_route – am_update_session_route
UE Policy Service	– reverse_ue_service – reverse_ue_service_notify
UDR	– udr_notify_route – reverse_user_udr_notify_service – reverse_user_udr_notify_service
CHF	– chf_terminate_route – chf_notify_route – reverse_user_chf_notify_service
NRF Client	– reverse_nrf_discovery_service – reverse_nrf_management_service – reverse_nrf_notify_service
SOAP Connector	– reverse_soapconnector_service

For details on configuring the route IDs, see *Route Level Configuration* section in *Oracle Communications Cloud Native Core, REST Specification Guide*.

Default Overload Threshold Values

This section describes the recommended default overload threshold values for Diameter Connector, PCRF Core, SM Service, AM Service, UE Policy Service, UDR Connector, and CHF Connector. To calculate threshold values, you must consider the resource values for microservices. The following table lists the default resource values for Diameter Connector, PCRF Core, SM Service, AM Service, UE Policy Service, UDR Connector, and CHF Connector:

Table 4-32 Default Resource Values

Resources	Diameter Connector	PCRF Core	SM Service	AM Service	UE Policy Service
CPU (Limits)	4	8	7	8	8
CPU (Requests)	3	7	7	7	7
Maximum CPU Usage	2.5	4.35	5.4	3.39	3.39
Maximum CPU Usage (%)	63	54	68	42	42
Maximum Replicas	8	8	8	8	8
Maximum TPS	2500	2100	880	675	300

Table 4-32 (Cont.) Default Resource Values

Resources	Diameter Connector	PCRF Core	SM Service	AM Service	UE Policy Service
Maximum TPS (all replicas)	20000	16800	7040	5400	2400
Worst RTT (assumed)	250 ms	250 ms	250 ms	250 ms	250 ms
Maximum Pending Transactions	5000	4200	1760	1350	600

Based on the values in Table 4-32 table, you can calculate the onset and abatement values for load levels - L1, L2, and L3, as shown in the Table 4-33 table.

> **Note**
>
> The values shown in Table 4-32 table are the default values and can be changed depending on the deployment model used.

Table 4-33 Formulas to Calculate Default Overload Threshold Levels

Load Level	CPU (%)	Pending Message Count (Absolute Value)	Failure Count (Absolute Value)
L1 - Onset	80% * C	60% * P	05% * T
L1 - Abatement	75% * C	50% * P	03% * T
L2 - Onset	90% * C	75% * P	10% * T
L2 - Abatement	85% * C	70% * P	08% * T
L3 - Onset	95% * C	90% * P	15% * T
L3 - Abatement	91% * C	85% * P	12% * T

Abatement value is the lower range where as the onset value is higher range for that particular level.

Note:

C denotes the maximum CPU utilization per pod of each of the services shown in Table 4-34 table.

T denotes the maximum incoming TPS of a given service listed in Table 4-34 table.

P denotes the maximum pending transaction size value based on worst RTT and maximum incoming TPS (T).

For example, if Maximum TPS (all replicas) (T) =20000 and Worst RTT (assumed) =250, then Maximum Pending Transactions (P) is calculated as (20000/1000) * 250 = 5000.

> **Note**
>
> You can configure the memory for each of these services.

Table 4-34 Messages to be accounted for calculating Service level Ingress TPS

Service Name	Message Types	Metrics Used
SM Service	• SM (Create, Update, Delete) • PA (Create, Update, Delete, events subscription, events un-subscription)	• ocpm_ingress_request_total • ocpm_ingress_response_total For more details on these metrics, see SM Service Metrics.
AM Service	AM (Create, Update, Delete)	• ocpm_ingress_request_total • ocpm_ingress_response_total For more details on these metrics, see AM Service Metrics.
UE Service	• UE (Create, Update, Delete) • N1 Message Notify	• ocpm_ingress_request_total • ocpm_ingress_response_total • ue_n1_transfer_request • ue_n1_transfer_response • ue_n1_transfer_ue_notification For more details on these metrics, see UE Service Metrics.

Table 4-34 (Cont.) Messages to be accounted for calculating Service level Ingress TPS

Service Name	Message Types	Metrics Used
Pcrf-Core	• Gx CCR (I/U/T) • Rx (AAR, STR)	• occnp_diam_request_local_total (CCR-I) • occnp_diam_response_local_total (CCA-I) • occnp_diam_request_local_total (CCR-U) • occnp_diam_response_local_total (CCA-U) • occnp_diam_request_local_total (CCR-T) • occnp_diam_response_local_total (CCA-T) • occnp_diam_request_local_total (AAR-I) • occnp_diam_response_local_total (AAA-I) • occnp_diam_request_local_total (AAR-U) • occnp_diam_response_local_total (AAA-U) • occnp_diam_request_local_total (RAR Gx) • occnp_diam_response_local_total (RAA Gx) • occnp_diam_request_local_total (RAR Rx) • occnp_diam_response_local_total (RAA Rx) • occnp_diam_request_local_total (STR) • occnp_diam_response_local_total (STA) For more details on these metrics, see PCRF Core Metrics.
Diameter Connector	• Rx (AAR, STR) • Sy SNR	• ocpm_ingress_request_total • ocpm_ingress_response_total For more details on these metrics, see Diameter Connector Service Metrics.
UDR Connector	Policy Data Change Notification	• ocpm_udr_tracking_request_total • ocpm_udr_tracking_response_total For more details on these metrics, see UDR Metrics.
CHF Connector	• Spending limit report notify • Subscription Termination	• ocpm_chf_tracking_request_total • ocpm_chf_tracking_response_total For more details on these metrics, see CHF Metrics.

The following table lists the default overload threshold values for Diameter Connector:

Table 4-35 Default Overload Threshold Values - Diameter Connector

Load Level	CPU (%)	Pending Message Count	Failure Count
L1 - Onset	51	3000	1000
L1 - Abatement	48	2500	600
L2 - Onset	57	3750	2000
L2 - Abatement	54	3500	1600
L3 - Onset	60	4500	3000
L3 - Abatement	58	4250	2400

The following table lists the default overload threshold values for PCRF Core:

Table 4-36 Default Overload Threshold Values - PCRF Core

Load Level	CPU (%)	Pending Message Count	Failure Count
L1 - Onset	44	2520	840
L1 - Abatement	41	2100	504
L2 - Onset	49	3150	1680
L2 - Abatement	47	2940	1344
L3 - Onset	52	3780	2520
L3 - Abatement	50	3570	2016

> **Note**
>
> On performing fresh installation of Policy, the system is loaded with only CPU threshold values. Using the sample configurations as a reference, you can set the threshold values for pending count and failure count according to the system's capacity.

The following table lists the default overload threshold values for UDR Connector:

Table 4-37 Default Overload Threshold Values - UDR Connector

Load Level	CPU (%)	Pending Message Count	Failure Count
L1 - Onset	51	1056	352
L1 - Abatement	54	880	208
L2 - Onset	61	1320	704
L2 - Abatement	57	1232	560
L3 - Onset	64	1584	1056
L3 - Abatement	62	1496	848

The following table lists the default overload threshold values for CHF Connector:

Table 4-38 Default Overload Threshold Values - CHF Connector

Load Level	CPU (%)	Pending Message Count	Failure Count
L1 - Onset	54	1056	352
L1 - Abatement	51	880	208
L2 - Onset	61	1320	704
L2 - Abatement	57	1232	560
L3 - Onset	64	1584	1056
L3 - Abatement	62	1496	848

Recommended Overload Threshold Values - SM, AM, and UE Services

The following table lists the default overload threshold values for SM, AM, and UE Services:

Table 4-39 Recommended Overload Threshold Values - SM, AM, and UE Services

Load Level	Service	CPU (%)	Pending Message Count	Failure Count
L1 - Onset	SM Service	55	1056	352
	AM Service	34	810	270
	UE Policy Service	34	360	120
L1 - Abatement	SM Service	51	880	212
	AM Service	32	675	162
	UE Policy Service	32	300	72
L2 - Onset	SM Service	61	1320	704
	AM Service	38	1013	540
	UE Policy Service	38	450	240
L2 - Abatement	SM Service	58	1232	564
	AM Service	36	945	432
	UE Policy Service	36	420	192
L3 - Onset	SM Service	65	1584	1056
	AM Service	40	1215	810
	UE Policy Service	40	540	360
L3 - Abatement	SM Service	62	1496	845
	AM Service	39	1148	648
	UE Policy Service	39	510	288

> **Note**
>
> These are the recommended values. It can be modified as per the customer requirements.
>
> The pending message count, and failure counts depend on the deployment capacity. The values shown in the above tables are calculated for the default deployment size. These values must be recalculated as per the deployment requirements.

Observe

Policy provides the following metrics specific to Overload Control feature:

- service_resource_stress
- service_resource_overload_level
- load_level
- system_overload_threshold_config_mode
- active_overload_threshold_fetch_failed

For more information, see PerfInfo Metrics section.

Alerts

- Policy provides the following alerts for overload control feature on SBI interface:
  - `ServiceOverloaded` - This alert is raised whenever a given service is in overload state - L1, L2, and L3.
  - `ServiceResourceOverLoaded` - This alert is raised when a given service is in overload state - L1, L2, or L3 due to resource types such as memory, CPU, pending count, and failure count.
- Policy provides `PERF_INFO_ACTIVE_OVERLOAD_THRESHOLD_FETCH_FAILED` alert for overload control threshold configuration. This alert is raised when the service is unable to fetch the current active overload threshold data.

Maintain

Error logs are generated when the system is overloaded and the actions taken to bring the system back to normal. Warning logs are generated to indicate the change in load level.

4.59.1 Overload Control- Diameter

For Diameter Gateway, Policy provides the following means for overload management:

- Pre-defined threshold load levels.
- Tracks number of pending messages of PCRF core and Diameter Connector from Diameter Gateway.
- Tracks CPU and memory usage of PCRF core and Diameter Connector.
- Enforce load shredding during various overload levels based on priority and percentage discard value for each priority. The priority and pecentage discard value are configurable.

Configure

To configure the threshold values (REST API only), discard priority, percentage discard value, and error codes for each defined level for overload control, you can use CNC Console as well as REST API.

- **Configure using CNC Console:** Perform the feature configurations on the Load Shedding Profiles and Message Priority Profiles page. For more information about the configurations, see Load Shedding Profiles.
- **Configure using REST API:** Policy provides the following REST API for configurating diameter gateway overload control feature:
  Load Shedding Profiles: {apiRoot}/oc-cnpolicy-configuration/v1/diameter/loadsheddingprofiles

Message Priority Profiles: {apiRoot}/oc-cnpolicy-configuration/v1/diameter/messageprioritypro files

You can perform the POST, PUT, or GET operations to configure the feature. For more information about REST API configuration, see *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

The following are the recommended configurations for load shedding profile and message priority profile respectively for overload control:

```
{
  "name": "default_overload_control_msg_priority_profile",
  "priorityRules": [{
    "ruleName": "Gx_CCR_I",
    "messagePriority": 14,
    "rulePriority": 1,
    "enableDRMPPriority": true,
    "conditions": {
      "application": "Gx",
      "message": "CCR",
      "preDefinedAVPConditions": [{
        "conditionName": "CC-Request-Type",
        "conditionCCRTValue": ["INITIAL_REQUEST"]
      }]
    }
  }, {
    "ruleName": "Rx_AAR_I",
    "messagePriority": 13,
    "rulePriority": 1,
    "enableDRMPPriority": true,
    "conditions": {
      "application": "Rx",
      "message": "AAR",
      "preDefinedAVPConditions": [{
        "conditionName": "Rx-Request-Type",
        "conditionRxRTValue": ["INITIAL_REQUEST"]
      }]
    }
  }, {
    "ruleName": "Gx_CCR_U",
    "messagePriority": 12,
    "rulePriority": 1,
    "enableDRMPPriority": true,
    "conditions": {
      "application": "Gx",
      "message": "AAR",
      "preDefinedAVPConditions": [{
        "conditionName": "CC-Request-Type",
        "conditionCCRTValue": ["UPDATE_REQUEST"]
      }]
    }
  }, {
    "ruleName": "Gx_CCR_T",
    "messagePriority": 8,
    "rulePriority": 1,
    "enableDRMPPriority": true,
    "conditions": {
```

```
      "application": "Gx",
      "message": "CCR",
      "preDefinedAVPConditions": [{
        "conditionName": "CC-Request-Type",
        "conditionCCRTValue": ["TERMINATION_REQUEST"]
      }]
    }
  }, {
    "ruleName": "Sy_SNR",
    "messagePriority": 6,
    "rulePriority": 1,
    "enableDRMPPriority": true,
    "conditions": {
      "application": "Sy",
      "message": "Sy-SNR",
      "preDefinedAVPConditions": []
    }
  }, {
    "ruleName": "Rx_STR",
    "messagePriority": 7,
    "rulePriority": 1,
    "enableDRMPPriority": true,
    "conditions": {
      "application": "Rx",
      "message": "STR",
      "preDefinedAVPConditions": []
    }
  }, {
    "ruleName": "Rx_AAR_U",
    "messagePriority": 11,
    "rulePriority": 1,
    "enableDRMPPriority": true,
    "conditions": {
      "application": "Rx",
      "message": "AAR",
      "preDefinedAVPConditions": [{
        "conditionName": "Rx-Request-Type",
        "conditionRxRTValue": ["UPDATE_REQUEST"]
      }]
    }
  }]
}


{
  "name": "default_overload_control_load_shedding_profile",
  "type": "Overload Control",
  "overloadLoadSheddingRules": [{
    "level": "L1",
    "discardPriority": 13,
    "ansWithResultCode": "DIAMETER_TOO_BUSY"
  }, {
    "level": "L2",
    "discardPriority": 11,
    "ansWithResultCode": "DIAMETER_TOO_BUSY"
  }, {
```

```
    "level": "L3",
    "discardPriority": 6,
    "ansWithResultCode": "DIAMETER_TOO_BUSY"
  }]
}
```

Observe

Policy provides the following metrics specific to Overload Control feature for diameter gateway:

- diam_overload_message_reject_total

For more information, see Diameter Gateway Metrics section.

4.59.2 Overload Control- SBI

For HTTP signaling, Policy overload management provides following means for overload management:

- Tracks number of pending messages for an exposed policy service in its NF Profile.
- Tracks number of failed responses (configurable as error code by operator) generated by exposed policy service in its NF Profile.
- Determine the overload level of system using data collected from point 1 and 2, against planned threshold levels (based on planning).
- Enforce load shredding during various overload levels.

Configure

To configure the discard policies, discard policy mapping, and error code profiles for overload control, you may use CNC Console as well as REST API.

> **Note**
>
> Currently, threshold values can be configured using REST API only.

- **Configure using CNC Console:** Perform the feature configurations on the Discard Policy Mapping, Discard Policy, and Error Code Profiles pages. For more information about the configurations, see Overload and Congestion Control Configurations.
- **Configure using REST API:** Policy provides the following REST API for configuring overload control feature on SBI interface:
  OC Policy Mapping: {apiRoot}/PCF/nf-common-component/v1/igw/ocpolicymapping

  OC Discard Policies: {apiRoot}/PCF/nf-common-component/v1/igw/ocdiscardpolicies

  Error Code Profiles: {apiRoot}/PCF/nf-common-component/v1/igw/errorcodeprofiles

  You can perform the GET, PUT, or PATCH operations to configure the feature. For more information about REST API configuration, see *Oracle Communications Cloud Natvie Core, Converged Policy REST Specification Guide*.

The following are the recommended configurations for default message priority values for overload control:

Table 4-40 Default Message Priority

Message Type	Priority
Create SM Policy Association	24
Delete SM Policy Association	16
Update SM Policy Association	18
UDR Notify	18
CHF Terminate	18
UDR Notify	18
CHF Notify	18
Create PA Application Session	24
Delete PA Application Session	16
Update PA Application Session	18
Subscribe PA Events	18
Unsubscribe PA Events	18
Create AM Policy Association	24
Delete AM Policy Association	16
Update AM Policy Association	18
Create UE Policy Association	24
Delete UE Policy Association	16
Update UE Policy Association	18

4.60 Load Shedding through Admission Control in PCRF-Core

If you are using Policy 22.4.5 or any other later versions, Load Shedding through Admission Control in PCRF-Core is enabled by default.

The Admission Control enhancement in PCRF-Core performs message load shedding during congestion. Load Shedding through the Admission Control feature is enabled by default in the PCRF-Core. This enhancement enables PCRF-Core to monitor the message backlog. When the number of outstanding messages to be processed exceeds a certain threshold for a certain duration, Admission Control starts rejecting a subset of the messages that it receives.

The default behavior when a node is too busy is to:

- reject Gx CCR-I requests at L1 level of busyness.
- reject Gx CCR-I and Rx AAR-I requests at L2 level of busyness.
- reject all the requests at level L3.

This prevents corresponding sessions to be established, as such if the load is due to these applications, the node matches the backlog and switches to the “busy” state.

Load Shedding through Admission Control in PCRF-Core allows for the definition of up to three levels of busyness based on the amount of backlog: level 1, level 2, and level 3. Level 1 is the least busy level and level 3 is the busiest level.

> **Note**
>
> The node is busy whenever it is at any of the busy levels.

Another way to look at the different busy levels is by the amount of processing time the node needs to catch up with the backlog. The “catch-up” processing time will translate into increased response latencies. So, as the busy level increases, more drastic measures may need to be taken to attempt to clear the backlog and go back to normal response times.

The entrance criteria for a busy level are:

- The backlog of outstanding messages in a node crossed a pre-defined threshold for the level.
- The backlog has been above the busyness level’s threshold for a minimum amount of time.

The time portion is added as hysteresis to avoid too frequent level transitions. When the thresholds for multiple levels are crossed, the level of busyness is determined by the highest crossed threshold, as long as the time criteria is satisfied.

> **Note**
>
> A node could bypass levels of busyness if it crosses multiple thresholds. As an example, a node could go from not being busy to being at busy level 2.

The exit criteria from a busyness level are:

- The backlog of outstanding messages has been smaller than a pre-defined threshold for the level.
- The backlog has been below the busyness level’s threshold for a minimum amount of time.

At each level, allow for the following actions per Diameter application, message type and AVPs content:

- Reject with a specific result code (default is DIAMETER_TOO_BUSY)
- Drop

For example, reject Gx CCR-I for APN X at level 1, reject Rx AAR-I message at level 2, and drop Gx CCR-U for APN Y at level 3.

The table below summarizes the default load shedding configuration values with which pcrf-core microservice is shipped with:

Table 4-41 Default Load Shedding Configuration

Busy Level	Entrance Threshold	Entrance Time Criterion (ms)	Exit Threshold	Exit Time Criterion (ms)	Actions
Level 1	300	300	150	500	• Reject Gx CCR-I with DIAMETER_TOO_BUSY • Accept CCR-I with DRMP=0 (Gx/CCR/CC-Request-Type=1 && DRMP=0)

Table 4-41 (Cont.) Default Load Shedding Configuration

Busy Level	Entrance Threshold	Entrance Time Criterion (ms)	Exit Threshold	Exit Time Criterion (ms)	Actions
Level 2	1000	300	600	500	• Reject Gx CCR-I with DIAMETER_TOO_BUSY • Reject Rx AAR-I with DIAMETER_TOO_BUSY • Accept CCR-I with DRMP=0 (Gx/CCR/CC-Request-Type=1 && DRMP=0)
Level 3	2000	300	1500	500	• DROP

For details on advanced configuration keys and values that are used for load shedding through admission control, see *PCRF-Core Configurations* section in *Cloud Native Core Policy Installation and Upgrade Guide*.

The following metric has been added in Diameter Gateway service for this feature:

- diameter_outstanding_msg_count

For more information, see Diameter Gateway Metrics.

> **Note**
>
> This metric can be used to create alerts as per the requirement.

4.61 Rate Limiting

With the support for rate limiting feature at Ingress Gateway, Ingress Gateway is expected to screen all configured routes and their respective rate limit configurations. Within the sampling period (configurable using REST APIs), Ingress Gateway calculates the rate for the required route along with the HTTP method. Then, it notifies the route level rate limiter rate about the calculated rate at the end of the sampling period. If the feature is enabled, any request with the sbi-priority header value greater than the configured value is discarded and Ingress Gateway returns the error response with configured errorCode.

> **Note**
>
> Ingress Gateway determines the number of messages to be silently dropped or rejected in the current sampling period on the basis of extra messages received in previous sampling period that are not rejected.

Managing Rate Limiting

Enable

Perform the following configurations to enable the rate limiting feature at Ingress Gateway:

- **CNC Console**: By default, this feature is disabled. To enable the rate limiting feature using CNC Console, set the **Enable Rate Limiting** parameter to `true` on the Rate Limiting Policy page.
- **REST API:** By default, this feature is disabled. To enable the rate limiting feature using REST API, set the `enabled` parameter to `true` in the following resource URI:

  Define rate limit: {apiRoot}/PCF/nf-common-component/v1/igw/routelevelratelimiting

  Define rate limit at route level: {apiRoot}/PCF/nf-common-component/v1/igw/routesconfiguration

  For more information, see the section Rate Limiting at Ingress Gateway in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

To configure the rate limiting policy, route level mapping, and error code profiles for rate limiting, you may use CNC Console as well as REST API.

- **Configure using CNC Console:** Perform the feature configurations on the Rate Limiting Policy, Route Level Mapping, and Error Code Profiles pages. For more information about the configurations, see Overload and Congestion Control Configurations.
- **Configure using REST API:** PCF provides the following REST API for configuring overload control feature on SBI interface:
  Define error code profiles: {apiRoot}/PCF/nf-common-component/v1/igw/errorcodeprofiles

  Define rate limit: {apiRoot}/PCF/nf-common-component/v1/igw/routelevelratelimiting

  Define rate limit at route level: {apiRoot}/PCF/nf-common-component/v1/igw/routesconfiguration

  You can perform the GET, PUT, or PATCH operations to configure the feature. For more information about REST API configuration, see *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

4.62 Topology Hiding for Diameter Gateway

With the Topolgy Hiding feature, it is possible to do the following:

- Hide topology-related information in messages sent to external peers.
- Restore the topology-related information in messages from external peers.

In Policy, the Diameter Gateway acts as a proxy for both incoming and outgoing diameter messages. Irrespective of the direction, request and response of diameter messages must include `Origin-Realm` and `Origin-Host` of diameter gateway on reaching external peers.

With Topology Hiding feature enabled, the Diameter Gateway verifies if the `Origin-Host` value in request or response from backend peers matches one of the managed backend peers, before sending to external peers. If yes, the value of `Origin-Host` is replaced with the Diameter Gateway identity in the message request and response. If the request or response is not for managed backend peers, no change is made to the `Origin-Host`.

Configure

To enable and configure this feature using CNC Console, the following parameters have been added under Diameter Configurations:

- **Topology Hiding**
- **Apps to Hide**

For more information about setting the parameter values, see Settings for Diameter Configurations.

Observe

Enabling the Topology Hiding feature replaces the Origin-Host dimension value with Diameter Gateway identity. It contains the value as per the message call-flow.

4.63 Two-phase Deployment of Policies

Policy allows you to configure and update the existing policies. However, updating or changing existing policies can impact existing traffic adversely. To overcome this challenge, Policy provides the two phase deployment model. This deployment model supports:

- Creation, modification, or deletion of policies, without impacting the existing policies.
- Testing of new policies with some percentage of live traffic before enabling it for 100% live traffic.

This feature allows you to create a policy project without deploying it. To implement this, Policy provides two possible states, **Prod** and **Dev** in the policy project.

By default, the **Dev** state is assigned to the policy project. Dev projects do not process any traffic in PRE (Policy Runtime Engine), where PRE is a pod. The Prod projects process traffic in PRE.

Managing Two-phase Deployment of Policies

Enable

The Two Phase deployment is a core functionality of Policy. You do not need to enable or disable this feature.

Configure

You can create new policy projects and update the existing projects using the CNC Console for Policy.

For more information about configuring Policy Projects, see Policy Projects.

4.64 Binding Mechanism Support (Nbsf)

PCF Session Binding is used for AF to discover a PCF that contains the PDU session and communicates with that PCF for App session procedures.

The PCF interacts with BSF to maintain the PCF session binding information during the PDU session life cycle.

PCF implements service operations as a consumer defined in clause 4.2 of TS 29.521[11]:

- Nbsf_Management_Register Service Operation

- Nbsf_Management_Deregister Service Operation

PCF invokes Nbsf_Management_Register Service Operation to the BSF to create PCF session binding when creating SM Policy Association. PCF invokes Nbsf_Management_Deregister Service Operation and Nbsf_Management_Register Service Operation when modify SM Policy Association and the UE address information is changed.

The system considers BSF selection based on the IP range to invoke BSF operation.

The interaction with BSF is optional and can be configured in PCF SM Service configuration. For more information about these configurations, see PCF Session Management.

Managing Binding Mechanism Function

Enable

You can enable the Binding mechanism functionality using the CNC Console or REST API for Policy.

- **Enable using CNC Console:** To enable the feature, set the **Binding Operation** parameter value to **TRUE** under the **Binding Configurations** group on the **PCF Session Management** page. For more information about enabling the feature through CNC Console, see PCF Session Management Service.
- **Enable using REST API:** Set the **bindingOperationEnabled** parameter value to **true** in the **Session Management Service** configuration API. For more information about enabling the feature through REST API, see "Session Management Service" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

You can configure the Binding mechanism functionality using the CNC Console or REST API for Policy.

- **Configure using CNC Console:** Perform the feature configurations under the **Binding Configurations** group on the **PCF Session Management** page. For more information about configuring the feature through CNC Console, see PCF Session Management Service.
- **Configure using REST API:** Policy provides the following REST API for Binding mechanism configuration:
  API: {apiRoot}/oc-cnpolicy-configuration/v1/services/pcfsm

  You can perform the PUT operation to configure the feature. For more information about REST API configuration, see Binding Configurations of the "Session Management Service" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Observe

Policy provides the Binding requests and response metrics in the Session Management services. For more information, see the Binding service metrics in SM Service Metrics.

4.65 LDAP Support

Policy establishes connections with data sources to retrieve information about subscribers from the database. It queries a data source using a key attribute that uniquely identifies a subscriber and stores the results in its cache. A data source uses this key attribute. For example, the phone or account number of the subscriber to index the information present in the database.

The Policy supports Lightweight Directory Access Protocol (LDAP) data source. It provides support for the following LDAP related functions:

- Configuring LDAP Server Information
- Providing support for multiple LDAP servers

Based on the conditions implemented in PCF system, Policy Data Source (PDS) retrieves all the relevant information from LDAP data source based on the rules configured in the system through LDAP gateway.

For information on configuring the LDAP server, see Data Source Configurations.

Managing LDAP

Enable

To enable the LDAP functionality, set value for the following parameter to true in the customer value file for Policy:

- **ldapGatewayEnable**

On setting the value for the aforementioned parameter to **true**, the Session Management (SM) or PCRF Core service routes the traffic to LDAP Gateway through PDS. For more information about setting the parameter values, see "Enabling/Disabling Services Configurations" in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide.*

Configure

LDAP credentails are stored as kubernetes secret along with Authentication DN and LDAP name. You must create a kuberenetes secret to store LDAP credentials before setting a PDS as LDAP data source.

To create a kubernetes secret for storing LDAP credentails:

1. Create a yaml file with the following syntax:

```
apiVersion: v1
kind: Secret
metadata:
  name: ldapsecret
  labels:
    type: ocpm.secret.ldap
type: Opaque
stringData:
  name: ldap1
  password: camiant
  authDn: uid=PolicyServer,ou=customer1,c=hu,o=customer1
```

   where, *name* is the configured LDAP server name.

   password is the LDAP credential for that data source.

   authDN is the authentication DN for that LDAP datsource.

> **Note**
>
> For different LDAP data sources, more entries can be added in above format only the key of the entry should be the ldap name specified in the Policy Graphical User Interface (GUI).

2. Create the kubernetes secret by executing the following command:

   ```
   kubectl apply -f yaml_file_name -n pcf-namespace
   ```

   where:

   yaml_file_name is a name of the yaml file that is created in Step 1.

   pcf-namespace is the deployment namespace used by the helm command.

Configure LDAP Server

If you are using LDAP for storing the user profiles, you must configure the LDAP server as the datasource. You can configure the LDAP server using the CNC Console or REST API for Policy.

- **Configure using CNC Console:** Perform the configurations for LDAP on the **Data Sources** page. For information on configuring the LDAP server, see Data Sources.
- **Configure using REST API:** Policy provides the following REST API for LDAP Server configuration:
  API: {apiRoot}/oc-cnpolicy-configuration/v1/datasources/{datasourceName}

  You can perform the PUT operation to configure the feature. For more information about REST API configuration, see *Data Source* in *Oracle Communications Cloud Native Core, Converged Policy REST API Specification Guide*.

After the configuration, you can use blockly action to retrieve resources from the LDAP datasource. For more information, see *PDS Category* in *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.

Observe

Policy provides metrics and alerts specific to LDAP Service. For information related to LDAP Gateway metrics, see LDAP Gateway.

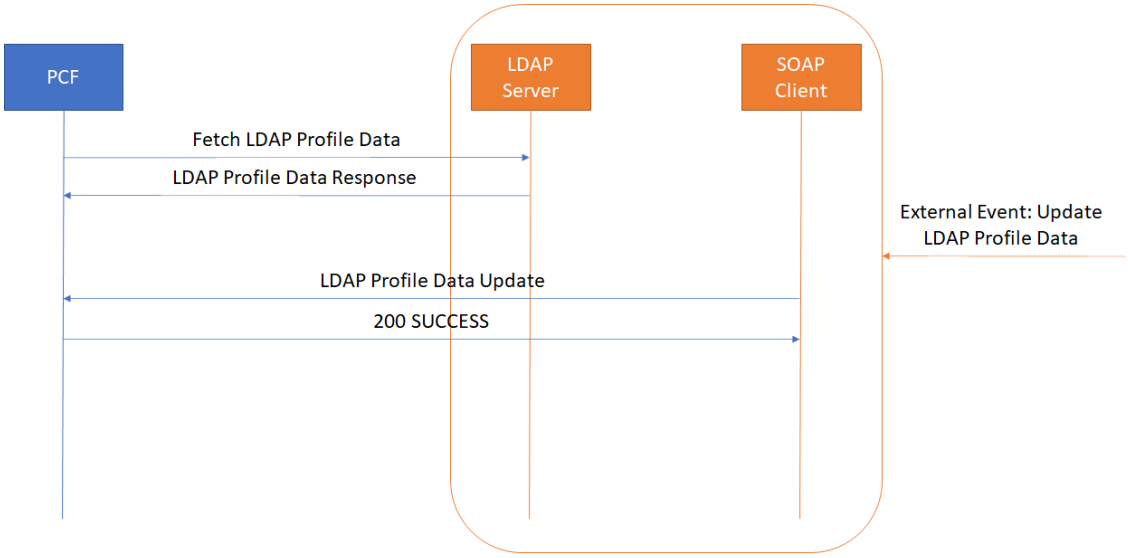
4.66 SOAP Support

Policy establishes connections with data sources to retrieve information about subscribers from the database. It queries a data source using a key attribute that uniquely identifies a subscriber and stores the results in its cache. A data source uses this key attribute. For example, the phone or account number of the subscriber to index the information present in the database.

Policy uses the subscriber data to:

- evaluate policy
- define FUP objects
- billing reset mechanism

Figure 4-99 Fetching subscriber data



1. When Policy recevies an SM Create request from SMF, it sends a request to the LDAP server to retrieve the subscriber profile.

   It uses IMSI, MSISDN, SUPI and GPSI to fetch the data.
2. Policy forwards the subscriber data received from LDAP server to PRE for evaluation.
3. Based on the outcome, Policy sends an SM Create response.to SMF.

Further SM Create request, Policy processes the subscriber profile that is cached. If the subscriber profile data is not available, Policy triggers an LDAP read to update the cache. Based on the evaluation of policy, it sends an SM Create Response.

The Policy uses SOAP (1.2 version) data interface to access subscriber data from an external LDAP Source.

1. Policy receives a notification from a SOAP-XML feed when the profile is updated.

   The SOAP notification message contains the changed LDAP attributes only along with the Subscriber Identifier (IMSI/MSISDN) identifying the affected subscriber. The SOAP notification is expected when at least one subscriber session is active on Policy.
2. Policy processes the notification, uses the subscriber identifier available as part of the SOAP notification to trigger additional LDAP search request.
3. It updates the subscriber profile database, and triggers a policy re-evaluation.
   It uses IMSI, MSISDN, SUPI, or GPSI to process the notification.
4. It then acknowledges the notification with a 200 ok response and discards the inactive sessions.

Managing SOAP

To enable the SOAP support, set value of `global.soapConnectorEnable` parameter to `true` in the custom_values.yaml file.

For more information about setting the parameter values, see *Enabling/Disabling Services Configurations* in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*.

4.67 Support for Timer Configurations

The timer profile sets request timeout values for UDR, CHF, BSF, and SMF messages in applications or interfaces. Policy applies the specific timeout profile while sending request to external NFs. The external NFs include UDR, CHF, BSF, and SMF. Policy updates the appropriate metrics in case of such request timeout.

In Policy, Ingress gateway applies operator configured interface specific timeout and use it as request timeout. It also propagate timeout value in `3gpp-Sbi-Max-Rsp-Time` header, so that it can be used by back-end services (if required).

Managing Timer Configurations

Enable or Configure
When back-end service has to originate a message for producer or consumer NF, the following algorithm for applying request timeout value is used:

- When the corresponding timer profile is not associated for the originating service:
  - If incoming request contains `3gpp-Sbi-Max-Rsp-Time` header, then use the value specified in `3gpp-Sbi-Max-Rsp-Time` header as timeout. For more information about configuring this value, see SBI Timer Header.
  - If the incoming request does not contain the `3gpp-Sbi-Max-Rsp-Time` header, then configure the request timeout during deployment or configuration using CNC Console for Policy. For more information about configuring Timer Profile using CNC Console, see Timer Profiles.
- When the corresponding timer profile is associated for the originating service:
  - If incoming request contains `3gpp-Sbi-Max-Rsp-Time` header, then use the minimum of the values specified in `3gpp-Sbi-Max-Rsp-Time` header and service or default timeout from the profile. For more information about configuring this value, see SBI Timer Header.
- If incoming request does not contain the `3gpp-Sbi-Max-Rsp-Time` header,
  - Use service specific timeout, if configured in timeout profile
  - Otherwise use default timeout from the timeout profile

After the above operations, the originating service also set the timeout value in the `3gpp-Sbi-Max-Rsp-Time` header to propagate it for any further producer services. Egress Gateway applies the request timeout from the `3gpp-Sbi-Max-Rsp-Time` header and finally uses its blocklisted feature to remove this header from egress message.

SBI Timer Header

SBI Timer Headers are implemented at Ingress Gateway. These are taken into consideration while calculating request timeout for a route along with route level (if configured) and global level request timeout.

Enable

To enable the SBI Timer Header, set the **ingress-gateway.isSbiTimerEnabled** parameter to **true**, under the **ingress-gateway** configurations in the custom-value.yaml file for Policy.

If the **ingress-gateway.isSbiTimerEnabled** parameter is set to **true**, then

- `3gpp-Sbi-Sender-Timestamp`, `3gpp-Sbi-Max-Rsp-Time`, and `3gpp-Sbi-Origination- Timestamp` parameters are used along with route level and global level request timeout to calculate the final request timeout.
- After calculating the final request timeout, original values of the `3gpp-Sbi-Sender-Timestamp`, `3gpp-Sbi-Max-Rsp-Time`, and `3gpp-Sbi-Origination- Timestamp` parameters are published in the `Orig-3gpp-Sbi-Sender-Timestamp`, `Orig-3gpp-Sbi- Max-Rsp-Time`, and `Orig-3gpp-Sbi-Origination-Timestamp` respective custom headers.

If the **ingress-gateway.isSbiTimerEnabled** parameter is set to **false**, then the SBI headers are not taken into consideration even if, they are present and the custom headers are not published.

4.68 Custom AVP to Support Third-Party Vendor Specific AVPs

An Attribute-Value Pair (AVP) is used to encapsulate protocol-specific information supported by CNC Policy. Diameter messages such as RAA, CCA, CCR, and RAR and so on, are supported by third-party AVP policy conditions. The supported outgoing Diameter messages set or remove third-party AVPs.

Policy allows you to create policy conditions to evaluate the presence of both standard (base) and third-party AVPs in Diameter messages or group AVPs during policy execution. A policy condition can check for the presence of both standard and third-party AVPs in incoming Diameter messages and evaluate their values. A policy action can use standard and third-party AVPs for routing, authentication, authorization, and accounting. Standard AVPs can be included in third-party AVP conditions and actions. To include a standard (base) AVP in a nonstandard application message, or to use a pre-standard AVP as a standard AVP, define it as a custom AVP.

Standard AVPs can be included in third-party AVP conditions and actions. To include a standard (base) AVP in a nonstandard application message, or to use a pre-standard AVP as a standard AVP, you must define it as a Custom AVP. When defined, custom AVPs are located at the end of a parent Diameter message or group AVP. If the parent AVP is null, the custom AVP is inserted at the root level of the message.

For example, a custom AVP definition appears at the end of this Charging-Rule-Install message:

```
 Charging-Rule-Install ::= < AVP Header: 1001 >
*[ Charging-Rule-Definition ]
*[ Charging-Rule-Name ]
*[ Charging-Rule-Base-Name ]
[ Bearer-Identifier ]
[ Rule-Activation-Time ]
[ Rule-Deactivation-Time ]
[ Resource-Allocation-Notification ]
[ Charging-Correlation-Indicator ]
*[ customAVP ]
```

A Set or Get SPR user attribute value can be set to the defined third-party AVP in Diameter messages. You can also set or remove the defined third-party AVPs during the execution point.

A third-party AVP is identified by a unique identifier in the following format:

name:*vendorId*

For example:

Condition
where the request *AVP NEW_AVP3:555* value is numerically *equal to 2012*

Parameters
The AVP name and vendor ID. In the example, the vendor ID is 555.

Description
A well-defined AVP custom name is referred to if the vendor ID is not specified.

When entering and sending a new third-party AVP definition to CNC Policy, the definition must include the AVP name, code, vendor ID, data type, and an optional AVP parameter.

Validation of the AVP code, Name, and vendor ID prohibits a user from overwriting the existing base AVPs.

These AVP actions include the ability to perform the following:

- Routing
- Authentication
- Authorization
- Accounting

For more information about custom AVP, see Custom AVP.

Managing Support for Custom AVP

Enable

This is a core functionality of Policy. This feature remains enabled by default.

Configure

You can add and configure the Custom AVPs using the CNC Console or REST API for Policy.

- **Configure using CNC Console:** Perform the configurations on the **Custom AVP** page. For more information about configuring PRAs, see Custom AVP.
- **Configure using REST API:** Policy provides the following REST API to create Custom AVPs:
  API: {apiRoot}/oc-cnpolicy-configuration/v1/policydata/common/customattributes/avp

  You can perform the POST, PUT, or GET operations to configure the Custom AVP. For more information about REST API configuration, see Custom AVP section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

After the configuration, you can use blockly actions to set values of Custom Values. For more information, see "PCRF-Core Actions" in *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.

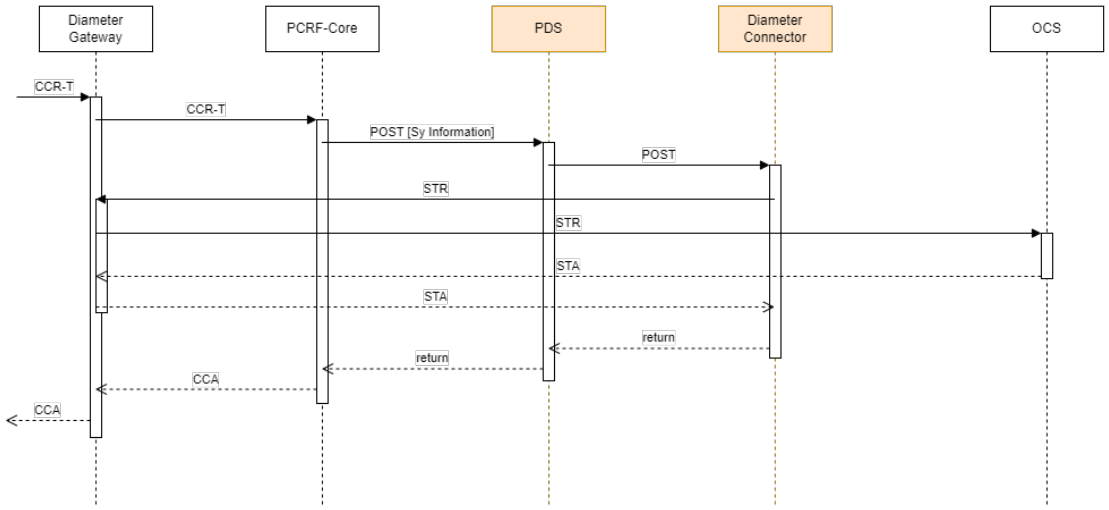
4.69 Adding Subscription-ID AVPs to STR messages

As part of the Final Spending Limit Report Request (SLR) procedure, PCRF sends a Session Termination Request (STR) to Online Charging System (OCS) through PDS and Diam-Connector to unsubscribe from all the Policy Counters belonging to the Diameter session and terminate the session.

OCS uses Subscription-ID AVPs to associate the STR with the SLR and close the session.

Without the Subscription-ID AVPs in STR, OCS returns a `DIAMETER_MISSING_AVP 5005` error to STRs.

Figure 4-100 Communicating STR messages



For this purpose, when PDS sends a GET request, the response header `oc-data-source-route-info` parameter from Diameter-Connector to PDS includes details such as `SusbscriptionIdType` and `UserIds` used in SLR from Diameter-Connector.

This same header is used by the PDS in the subsequent OCS request including INTERMEDIATE and TERMINATE requests.

In case of a TERMINATE request, the Diameter-Connector receives the `oc-data-source-route-info` header from PDS. It extracts `SusbscriptionIdType` and `UserIds` if `OCS_ENABLE_SUBS_ID_ON_STR` is set to `true`.

It then sends the Subscription-Id AVP in STR message.

Managing the Adding Subscription-ID AVPs to STR messages feature

Enable

You can enable addition of Subscription-Id information in Subscription-Id AVPs using `diam-connector.envSyEnableSubsIdOnSTR` Helm parameter during installation. For more information, see *Diameter Gateway and Diameter Connector Configuration* section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.

Configure

You can configure addition of subscription-Ids using the `diam-connector.envSyEnableSubsIdOnSTR` Helm parameter during installation.

For more information, see *Diameter Gateway and Diameter Connector Configuration* section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.

4.70 Asynchronized Access to nCHF

The Policy solution supports asynchronized access of subscriber related information from data sources such as CHF.

Managing Asynchronized Access to nCHF

Enable

You can enable the Asynchronized Access to nCHF functionality using the CNC Console or REST API for Policy.

- Enable the **Enable Async CHF Query** parameter under the **User** group on the **PCF Session Management** page. For more information, see information about user configurations in PCF Session Management.
- **Enable using REST API:** Set the **enableChfQueryAll** parameter value to **true** in **User Configuration** of the **Session Management Service** API. For more information about enabling the feature through REST API, see "Session Management Service" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

After enabling this feature, you can perform the necessary configurations using the **Fetch Policy Counters from CHF** block for the SM service. Configure this block to fetch policy counters through asynchronous CHF lookup.

For more information about configuring the block, see "Public Category" in *Oracle Communications Cloud Native Core, Converged Policy Design Guide.*

4.71 Nudr to Support OperatorSpecificData

Policy supports custom JSON feature that allows you to configure flexible data schema as part of OperatorSpecificData for policy validation from Nudr interface and apply policies based on the data set. The advantages of this integration are:

- Uses new data type object for data not mapped to standardized fields
- Object contains IDs and map of Operator specific policy set objects (blobs)
  Blobs can be used to store boolean, number, object, array, null, string *<JSON, XML, Base64 encoded binary data>*, and so on.
- Aligns with 3GPP

Managing Nudr Support for OperatorSpecificData

Enable

You can enable the Nudr Support for OperatorSpecificData functionality using the CNC Console for Policy.

Enable the **Enable Operator Specific Data Query** parameter under the **User** group on the **PCF Session Management** page. For more information about enabling the feature through CNC Console, see PCF Session Management.

Configure
You can use tThe OperatorSpecificData as an input in different Policy Projects. For more information, see *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.

4.72 Support for Spending Limit Pending Counter

The Policy solution enables you to use activation time in the pending policy counter as input to the Condition Data for either activation time or deactivation time. This functionality is available for condition data used by PCC rules and session rules.

> **Note**
>
> This functionality is only supported by Session Management (SM) Associations.

Example:

The following is a sample of Spending Limit Pending Counter:

```
{
 "supi": "imsi-450081100100001",
 "gpsi": "msisdn-9192503899",
 "PolicyCounterIds": [
 {
 "policyCounterId": "silver",
 "currentStatus": "valid",
 "penPolCounterStatuses": [
 {
 "policyCounterStatus": "start",
 "activationTime": "2020-05-16T16:25:00.659Z"
 }
 ]
 },
 {
 "policyCounterId": "gold",
 "currentStatus": "valid",
 "penPolCounterStatuses": [
 {
 "policyCounterStatus": "start",
 "activationTime": "2020-01-01T00:00:00.000Z"
 },
 {
 "policyCounterStatus": "end",
 "activationTime": "2020-12-31T23:59:59.000Z"
 }
 ]
 }
 ]
}
```

In the above sample, PCF selects one of the policy counters by `policyCounterId` and `policyCounterStatus`. After this, PCF uses the `activationTime` as the activation time or deactivation time for the condition data of PCC Rule or Session Rule, and finally sends the related PCC Rule or Session Rule to SMF.

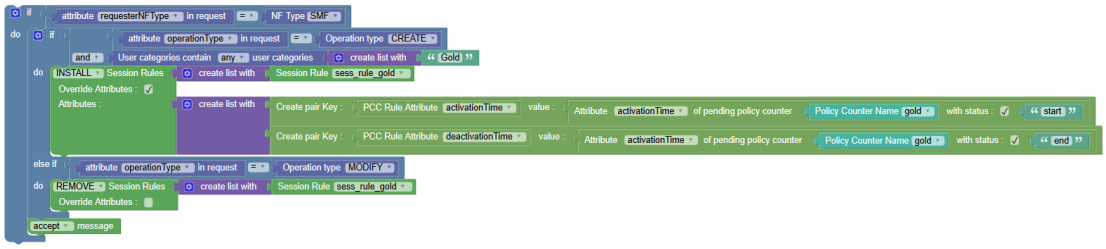
The existing policies support the spending limit pending counter functionality. You can perform the following configurations:

- Select the activationTime of pending counter by policyCounterId and policyCounterStatus.
- Install PCC Rule or Apply PCC Rule Profile using the activationTime as activationTime or/and deactivationTime of Condition Data of the PCC Rule or PCC Rule Profile.
- Install/Modify Session Rule using the activationTime as activationTime or/and deactivationTime of Condition Data of the Session Rule

For more information on Activation/Deactivation Time of PCC Rules/Session Rules blockly design for policy design, see **PCF-SM Category** section in *Oracle Communcations Cloud Native Core, Converged Policy Design Guide*.

A Policy Sample, this sample is used to install session rule using policy pending counter from the above sample.

Installing Session Rule Using Policy Pending Counter



Managing Spending Limit Pending Counter Support

Enable
This feature remains enabled by default.

Configure
You can configure the Spending Limit Pending Counter functionality by configuring the following SM service functions:

- **Policy Counter ID**: Perform the Policy Counter Id configurations on the **Policy Counter ID** page. For more information, see Policy Counter Id,
- **Condition Data**: Perform the Condition Data configurations on the **Condition Data** page. For more information, see Condition Data.
- **PCC Rule**: Perform the PCC Rule configurations on the **PCC Rule** page. For more information, see PCC Rule.
- **Session Rule**: Perform the Session Rule configurations on the **Session Rule** page. For more information, see Session Rule.

After the above configurations, you can use blockly action to install session rule/PCC rule using policy pending counter . For more information, see "PCF-SM Category" in *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.

4.73 XFCC Header Validation

Overview

With XFCC Header Validation feature, Policy (PCF) as a producer, checks if the SCP that is sending the HTTP request is the same SCP that is configured in the PCF. Policy performs this

check by comparing the FQDN of the SCP present in the “x-forwarded-client-cert” (XFCC) of http2 header with the list of FQDN of the SCPs configured in the PCF. This configured list contains all the host FQDNs resolved successfully via DNS-SRV as well as static SCPs. The header validation can be enabled at global as well as at the route level.

> **Note**
>
> This feature is applicable only when SCP is deployed in the network topology.

XFCC header validation applies to all the interfaces of Policy (PCF), which support indirect communication through SCP. The list of the interfaces is as follows:

- AM, SMF, and AF SBI service requests and responses
- UDR, CHF, and NRF notification messages

Configuring SCPs at Policy

To configure SCP, you need to customize `custom-values.yaml` at the time of deploying Policy.

In the earlier releases, users could only configure SCPs statitcally as shown in the following snippet:

```
xfccHeaderValidation:
    validation:

      enabled: false
      nfList:
        - scp.com
        - smf.com
        - amf.com
```

Starting Policy release 22.1.0, you can configure single or multiple virtual FQDNs for the SCP along with the static configuration as shown in the following snippet:

```
global:
    xfccHeaderValidation:
      validation:
        enabled: false
        peerList:
          - name: scp.com
          - name: smf.com
          - name: amf.com
          - name: scp1.com
            enabled: true
          - name: scp2.com
          - name: scp3.com
            enabled: false
          - name: xyz.test.com
            enabled: true
            scheme: http
            type: virtual
          - name: abc.test.com
            enabled: true
```

```
    scheme: https
    type: virtual
  - name: xfcc.test.com
    enabled: false
    scheme: http
    type: virtual
```

Static SCP: To define an SCP instance statically, add the name and set enabled parameter to true in the peerList. If the enabled parameter is set to false for an instance, then it is not included in the list of configured FQDNs. If you do not specify enabled parameter then by default it is considered as true.

Virtual SCP: To define an SCP with virtual FQDN, add the name, scheme as http or https, type as virtual, and set enabled parameter to true. If the enabled parameter is set to false for an instance, then it is not included in the list of configured FQDNs.

Resolving FQDNs to find Authorized SCPs for Policy

During the bootup of Ingress Gateway, it tries to resolve the configured virtual FQDN via Alternate Route service using the following helm configuration:

```
dnsSrv:
    port: *svcAlternateRouteServiceHttp #Alternate-route port for scheme
'http'. Change is required if the scheme below changes.
    scheme: http
```

If Alternate Route service is unable to resolve the configured virtual host, Ingress Gateway stores it in the list of failed FQDNs and reattempts the request at 300 s (default value configured for **dnsResolutionInterval**).

The following metric is used when the request to resolve configured virtual FQDNs is unsuccessful:

- `oc_ingressgateway_dns_resolution`: This metric is pegged when DNS resolution for a given FQDN fails.
- `oc_ingressgateway_dns_resolution_failure`: This is a gauge metric that is triggered when DNS resolution for a given FQDN fails.

Handling Traffic Flow

The specification for the XFCC header validation of an incoming request with the FQDN's of SCP configured at the IGW are as follows:

1. When a request is received at IGW, it gets forwarded to the back-end micro-service based on the configured rules.
2. The **ingress-gateway.xfccHeaderValidation.validation.matchField** parameter configured in IGW remains available in the client certificate field of XFCC header. You must validate this parameter with the configured FQDNs of SCP.
3. If multiple XFCC headers are present in the incoming request to IGW, then the IGW validates the **ingress-gateway.xfccHeaderValidation.validation.matchField** parameter present in each XFCC header against the configured FQDNs of SCP, until a match is found.

There are following scenarios to handle the XFCC header present in the incoming request to IGW:

- Validation of single XFCC header present in the incoming request to IGW

- Validation of multiple XFCC headers present in the incoming request to IGW

Validating single XFCC Header
The following figure describes the call flow for validation of single XFCC header:

Figure 4-101 Call Flow Validating single XFCC Header

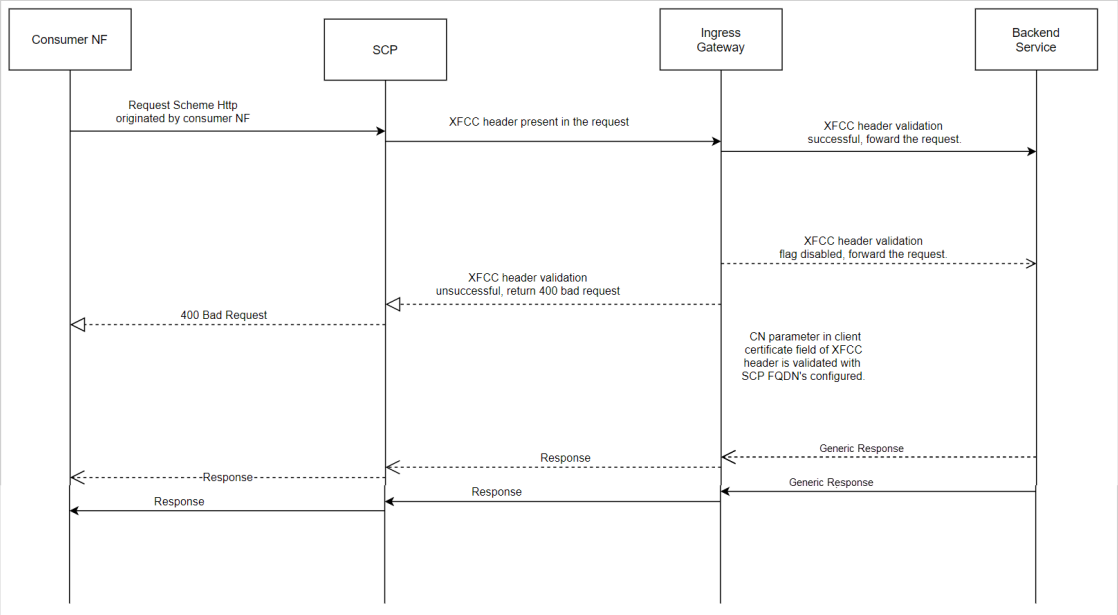


Table 4-42 Single XFCC Header Configuration

Prerequisite	Condition	Steps
Given XFCC Header validation parameter is enabled for the corresponding route match and matchCerts count correctly configured.	If the matchField parameter of client certificate field in XFCC header matches with one of the FQDNs of SCP configured.	Forward the request to back-end microservice and receive a corresponding response.
Given XFCC header validation parameter is enabled for the corresponding route match and matchCerts count correctly configured.	If matchField parameter of client certificate field in XFCC header does not match with the FQDNs of SCP configured.	Return 400 Bad Request response from Ingress Gateway.
Given XFCC header validation parameter is not enabled for the corresponding route match.		Forward the request to the back-end microservice and receive a corresponding response.

Example of XFCC Header:

```
x-forwarded-client-cert: By=http://
router1.blr.com;Hash=468ed33be74eee6556d90c0149c1309e9ba61d6425303443c0748a02dd8d
e68; Subject="/C=US/ST=CA/L=San Francisco/OU=Lyft/CN=Test Client"; URI=http://
testenv1.blr.com;DNS=blr.com; DNS=www.blr.com
```

Validating multiple XFCC Headers
The following figure describes the call flow for validation of multiple XFCC header:

Figure 4-102 Call Flow Validating Multiple XFCC Headers

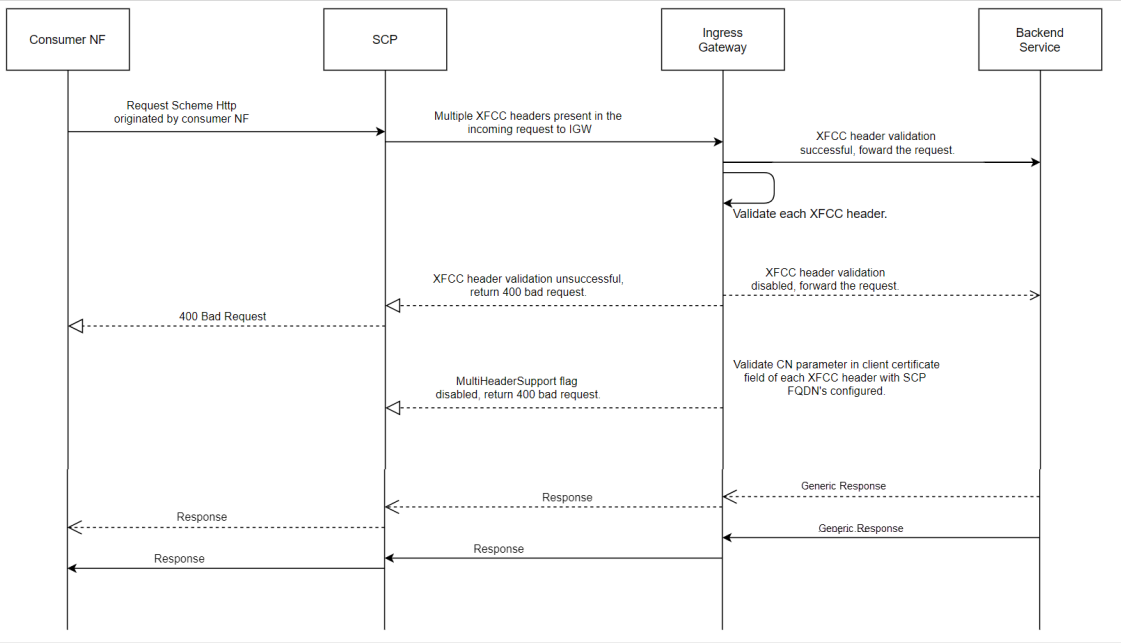


Table 4-43 Multiple XFCC Headers Configuration

Prerequisite	Condition	Steps
Given XFCC header validation parameter is enabled and matchCerts count correctly configured to validate across XFCC header certificates from the right most entry.	If matchField parameter of the corresponding client certificate field being validated against currently in the corresponding XFCC header matches with the FQDNs of SCP configured at Ingress Gateway.	Consider the request as a valid request and forward the request to the back-end micro-service and receive a corresponding response.
Given XFCC header validation parameter is enabled and matchCerts count correctly configured to validate across XFCC header certificates from the right most entry.	If matchField parameter of client certificate field in corresponding XFCC headers do not match with the FQDNs of SCP configured at Ingress Gateway for the corresponding matchCerts count.	Consider the request as an invalid request and return a 400 Bad Request response from IGW. For more information about error codes, see SBI Ingress Error Code Profiles Collection.
Given XFCC header validation parameter is enabled and matchCerts count -1.		Consider the request as valid request and match against the corresponding match field in all XFCC headers, if validation successful then forward the request else return 400 BAD Request.
Given XFCC header validation parameter is not enabled.		Forward the request to back-end microservice and receive a corresponding response.

Example of XFCC Header:

```
x-forwarded-client-cert:By=http://
router1.blr.com;Hash=468ed33be74eee6556d90c0149c1309e9ba61d6425303443c0748a02dd8d
```

```
e68; Subject="/C=US/ST=CA/L=San Francisco/OU=Lyft/CN=nf1.com"; URI=http://
testenv1.blr.com; DNS=nf8.com; DNS=nf1.com; DNS=nf6.com, By=http://
router1.blr.com;Hash=468ed33be74eee6556d90c0149c1309e9ba61d6425303443c0748a02dd8d
e68; Subject="/C=US/ST=CA/L=San Francisco/OU=Lyft/CN=nf10.com"; URI=http://
testenv1.blr.com; DNS=nf10.com; DNS=nf8.com; DNS=nf9.com, By=http://
routexr1.blr.com;Hash=468ed33be74eee6556d90c0149c1309e9ba61d6425303443c0748a02dd8
de68; Subject="/C=US/ST=CA/L=San Francisco/OU=Lyft/CN=nf4.com"; URI=http://
testenv1.blr.com; DNS=nf9.com; DNS=nf4.com;DNS=nf1.com
```

Managing XFCC Header Validation

Enable

- **Global Level**: To enable or disable the XFCC header validation feature, set the value of the `ingress-gateway.global.xfccHeaderValidation.validation.enabled` to true or false respectively.
- **Route Level**: To enable or disable the XFCC header validation feature at route level, set the value of the `xfccHeaderValidation.validationEnabled` under routesConfig to true or false respectively.

> **Note**
>
> If the `xfccHeaderValidation.validationEnabled` parameter is defined at route level, then the configuration takes precedence over global configuration.

For instance, if you want to enable XFCC header validation for selected routes, then set the global parameter as false and make route specific configuration to true.

```
global:
    xfccHeaderValidation:
      validation:
        enabled: false


routesConfig:
    - id: sm_create_session_route
      uri: http://{{ .Release.Name }}-occnp-pcf-sm:
{{ .Values.global.servicePorts.pcfSmServiceHttp }}
      path: /npcf-smpolicycontrol/*/sm-policies
      order: 1
      method: POST
      readBodyForLog: true
      filters:
        subLog: true,CREATE,SM
      metadata:
        xfccHeaderValidation:
          validationEnabled: true
```

For more information about setting the parameter values, see "XFCC Header Validation Configuration" section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*.

Configure

You can configure the various parameters related to XFCC Header Validation feature in the `custom-values.yaml`. For more information, see "XFCC Header Validation Configuration" section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*.

Observe

The XFCC Header Validation feature can be observed using the metrics specific to Ingress Gateway. For information related to Ingress Gateway metrics, see Ingress Gateway Metrics.

Configuring Error Codes

When the XFCC header validation feature is enabled and SCP FQDN in the incoming header does not match the configured FQDN in PCF, XFCC header is not present, or XFCC header is invalid, then PCF may return error in the response. Users have the ability to customize the error code returned in the response using the following helm configuration:

```
errorTrigger:
          - exceptionType: XFCC_HEADER_INVALID
            errorCode: '401'
            errorCause: xfcc header is invalid
            errorTitle: 'Invalid XFCC Header'
            errorDescription: 'Invalid XFCC Header'
          - exceptionType: XFCC_MATCHCERTCOUNT_GREATER_THAN_CERTS_IN_HEADER
            errorCode: '402'
            errorCause: matchCerts count is greater than the certs in the
request
            errorTitle: ''
            errorDescription: ''
          - exceptionType: XFCC_HEADER_NOT_PRESENT_OR_EMPTY
            errorCode: '403'
            errorCause: xfcc header is not present or empty in the request
            errorTitle: ''
            errorDescription: ''
```

If the configured error code in the errorCodeOnValidationFailure field lies in 3xx error series only then the values for retryAfter and redirectUrl if configured under XFCC Header Validation Configuration at Ingress Gateway are used to populate Retry-After and LOCATION headers correspondingly while sending error response from Ingress Gateway.

4.74 Policy Event Records

The Policy Event Records feature enables you to view transaction related information that includes input data set, policies that were evaluated, and resulting policy actions from any transactions. Policy event records provides an insight on the policy flow, conditions that are executed, and policy actions enforced by the Policy solution. Single record is generated including the following information:

- Transaction and input Data input set
- Executed policies
- Flow of the executed policies
- Executed instructions or rules
- Policy results and actions

The format of this data is in accordance with the JSON format.

Following are the attributes of the JSON schema:

- policyStartTime
- policyRequest
- policyExecution
- policyEndTime
- policyResponse

Sample Schema:

```
{
    "policyStartTime": "2021-09-14T09:00:37.261Z",
    "policyRequest": {
      "request": {
        "requestType": "SMF",
        "operationType": "CREATE"
      }
    },
    "policyExecution": "[\" Start evaluating policy
main\",\"utils.getAttributeVal(['request.request.operationType'], request) ==
'CREATE' evaluates to be true\",\" INSTALL  Session Rules
     ['sessionRule2']\",\" End evaluating policy main\"]",
    "policyEndTime": "2021-09-14T09:00:37.262Z",
    "policyResponse": {
      "actions": [],
      "customAttributeActions": [],
      "variables": {},
      "ruleActions": [
        {
          "actionType": "INSTALL",
          "ruleType": "SESSION_RULE",
          "sessionRule": [
            {
              "sessRuleId": "sessionRule2"
            }
          ]
        }
      ]
    }
  }
```

For more information on configuration parameters related to policy event records, see General Settings section.

4.75 URSP Policy Support

With this enhancement, you should be able to define URSP policy in PCF and deliver the policy to UE. UE Route Selection Policy (URSP) is used by the UE to determine how to route outgoing traffic. Traffic can be routed to an established PDU Session, can be offloaded to non-3GPP access outside a PDU Session, or can trigger the establishment of a new PDU Session.

- UE policy service should be able to establish UE Policy Association requested by the NF service consumer
- UE policy service should be able to define and deliver URSP message to UE via AMF using N1N2 message

For more information about configuring URSP, see URSP Rule.

Managing URSP

Enable

This is a core functionality of Policy. It remains enabled by default.

Configure

You can configure the URSP Rules using the CNC Console or REST API for Policy.

- **Configure using CNC Console:** Perform the feature configurations on the **URSP Rule** page. For more information on configuring the feature through CNC Console, see URSP Rule.
- **Configure using REST API:** Policy provides the following REST API for URSP configuration:
  API: {apiRoot}/oc-cnpolicy-configuration/v1/policydata/pcfue/ursps

  You can perform the POST and PUT operation to configure the feature. For more information on REST API configuration, see "URSP" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Observe

Policy provides the UE service metrics for observing all the features related to the service. For more information, see the UE Service metrics in UE Service Metrics.

4.76 AF Integration over Rx

Rx interface handling supports the following:

- Diameter session management (creation, modification and deletion)
- Session association and trigger policies on N7 interface with PCC rules
- Compliant with 29.214 V15.1.0
- Integration with BSF for session binding management
- Integration with UDR for updating policyInfo and SM policy association

4.77 IMS Emergency Session Support

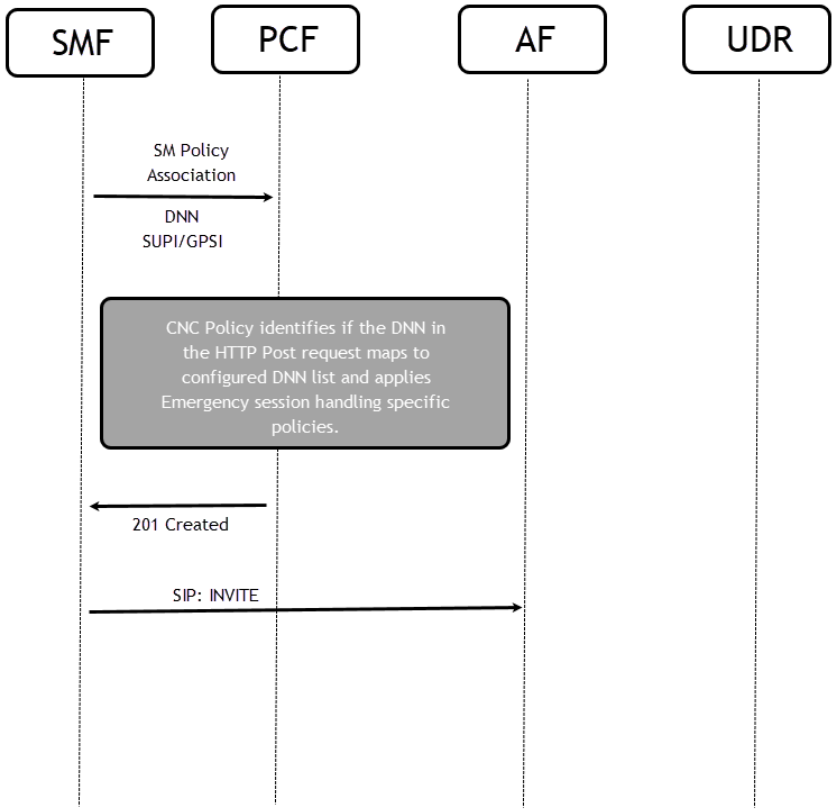
This feature allows a caller to contact local eCNC Policymergency services, such as Police, Fire departement, or ambulance for assistance. To ensure uninterrupted services, separate Data Network Names (DNNs) are defined to support the emergency services sessions. The subscription related information is not relevant for such sessions.

The IMS Emergency Support provides the folowing capabilities:

- – All the devices, with or without SIM can trigger an emergency session and provided access to the emergency services
  – The local operator policies are applied for QoS determination

The following figure describes the call flow for the Emergency Session handling in Policy:

Figure 4-103 Call Flow - IMS Emergency



- SMF sends an HTTP POST message for establishing SM Policy Association and the "dnn" attribute including the Emergency DNN. The SMF includes the SUPI within the "supi" attribute and the GPSI if available within the "gpsi" attribute.
- The PCF detects that the PDU session is restricted to IMS Emergency services when the HTTP POST message is received and the "dnn" attribute includes a data network identifier that matches one of the Emergency DNs from the configurable list.
- Then, suitable PCC Rules are applied restricting the access to Emergency Services in a response message.

> **Note**
>
> Policy does not support the IMS restoration for emergency sessions.

Managing IMS Emergency

Enable

This feature remains enabled by default.

Configure

You can configure the IMS Emergency functionality using the CNC Console or REST API for Policy.

- **Configure using CNC Console:**
  - To configure emergency DNN which allows for N7 request to be successful without MSISDN or IMSI, perform the feature configurations under the **IMS Emergency Session** group on the **PCF Session Management** page. For more information, see PCF Session Management Service.
  - To configure emergency DNN for Rx calls, perform the feature configurations under the **IMS Emergency Session** group on the **PCF Policy Authorization** page. For more information, see PCF Policy Authorization.
- **Configure using REST API:** Policy provides the following REST API for Binding mechanism configuration:
  - To configure emergency DNN which allows for N7 request to be successful without MSISDN or IMSI:
    API: {apiRoot}/oc-cnpolicy-configuration/v1/services/pcfsm

    You can perform the PUT operation to configure the feature. For more information on REST API configuration, see **IMS Emergency Session Configuration** of the "Session Management Service" in *Oracle Communications Cloud Native Core, Converged Policy REST API Specification Guide*.
  - To configure eergency DNN for Rx calls:
    API: {apiRoot}/oc-cnpolicy-configuration/v1/services/pcfpa

    You can perform the PUT operation to configure the feature. For more information on REST API configuration, see **IMS Emergency Session Configuration** of the "Policy Authorization Service" in *Oracle Communications Cloud Native Core, Converged Policy REST API Specification Guide*.

After the configuration, you can use the blockly actions for IMS Emergency related action. For more information, see "IMS Emergency Session Support" in *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.

Observe

Policy provides metrics and alerts specific to SM service and Policy Authorization service.

For information related to SM service metrics, see SM Service Metrics.

For information related to Policy Authorization service metrics, see PA Service Metrics.

4.78 IMS Restoration

Policy supports restoring an IMS network as defined in 3GPP TS 29.212V15.4.0.

Also, it supports enhancements to the restoration procedure as defined in 3GPP TS 29.214V15.4.0.

> **Note**
>
> This feature is applicable only for PCRF flows in Converged mode of deployment.

The automatic restoration procedure involves:

1. If AF provisions information about the AF signalling flows between the UE and the AF, PCRF installs the corresponding dynamic PCC rules by triggering a RAR message.
2. The PCRF provides the Charging-Rule-Install AVP including the Charging-Rule-Definition AVP(s).

   The Charging-Rule-Definition AVPs include:

   - the flow-Information AVP, the signalling flows between UE and the AF.
   - AF-SignallingProtocol AVP set to the value corresponding to the signalling protocol used between the UE and the AF.
3. PCRF shares the AF address with PCEF.
4. PCEF:
   a. Responds to PCEF with a RAA message.
   b. Initiates the corresponding bearer procedure if required.
   c. Extracts the AF address from the PCC rules.
   d. Checks if the monitoring procedure must be started for the corresponding AF.
   e. Starts the monitoring procedure as defined for the different access types.

In case AF de-provisions the information about the AF signalling flows between the UE and the AF, as defined in 3GPP TS 29.214 [10] Section 4.4.5a:

1. PCRF removes the corresponding dynamic PCC rules by triggering a RAR message.
2. PCRF sends the Charging-Rule-Remove AVP including the corresponding Charging-RuleName AVP(s) to PCEF.
3. PCEF uses the AF address associated with the removed rule to check if it can stop monitoring the corresponding AF.
4. PCEF acknowledges by sending a RAA command to the PCRF.

In case of P-CSCF restoration enhancement,

1. PCRF receives a request for P-CSCF restoration from the P-CSCF.
2. PCRF sends a Gx RAR command including the PCSCF-Restoration-Indication AVP set to value 0 (PCSCF_RESTORATION) to the PCEF for the corresponding Gx session.
3. PCEF acknowledges the RAR command by sending an RAA command to the PCRF and initiates the corresponding bearer procedure for the IMS PDN connection as defined in 3GPP TS 23.380 [33].

4.79 Automated Test Suite Support

Policy provides Automated Test Suite (ATS) for validating different functionalities. ATS allows you to execute Policy test cases using an automated testing tool, and then compares the actual results with the expected or predicted results. In this process, there is no intervention from the user is required. For more information, see *Oracle Communications Cloud Native Core, ATS User Guide*.

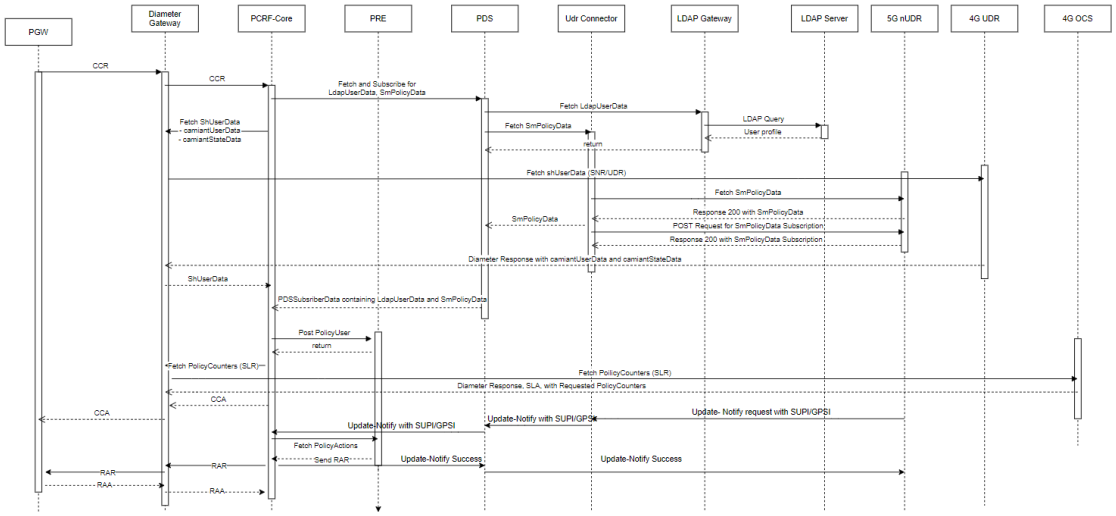
4.80 Notification Handling from PDS for PCRF-Core

In earlier releases, PCRF-Core could communicate with 4G UDR and OCS (Sy interface) through Diameter Gateway. PCRF-Core could communicate with LDAP through PDS. With the

latest enhancements, PCRF-Core can now communicate with 5G nUDR through PDS to fetch SmPolicyData.

As illustrated in the following call flow diagram, when PCRF-Core receives notify request with SUPI or GPSI from PDS for smPolicyData, that is, from nUDR - PCRF-Core initiates reauthorization request, and populates PolicyUser structure. Then, it sends the evaluation request to PRE. PRE further evaluates policy based on the new structure received. If installed rules or any of its related fields are different from the PRE Evaluation result, then PCRF-Core triggers a RAR request towards PGW.

Figure 4-104 Call flow for notification handling from PDS for PCRF-Core



Enable

To enable this feature, set the values for the following fields to **true**, on Settings page:

- **SmPolicyData** (Enable)
- **Subscribe to Notify** (SmPolicyData Attributes)

4.81 Service mesh for intra-NF Communication

Policy leverages the Istio or Envoy service mesh (Aspen Service Mesh) for all internal and external communication. The service mesh integration provides inter-NF communication and allows API gateway co-working with service mesh. The service mesh integration supports the services by deploying a special sidecar proxy in the environment to intercept all network communication between microservices.

The Aspen Service Mesh (ASM) configurations are classified into:

- **Control Plane**: It involves adding labels or annotations to inject sidecar.
- **Data Plane**: It helps in traffic management like handling NF call flows by adding Service Entries (SE), Destination Rules (DR), Envoy Filters (EF) and other resource changes like apiVersion change between versions. This is done manually depending on each NF requirement and ASM deployment.

Managing Service mesh for intra-NF Communication

Enable

To enable Aspen Service Mesh, configure the following parameters under **nrf-client-nfdiscovery**, **ingress-gateway**, **egress-gateway**, and **alternate-route** sections in the custom values file for Policy:

- serviceMeshCheck
- istioSidecarQuitUrl
- istioSidecarReadyUrl

For more information on enabling the parameter value, see "Aspen Service Mesh Configurations" in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide.*

Configure
The Aspen Service Mesh (ASM) configurations are classified into:

- **Control Plane**: For information on configuring the parameter values, see "Aspen Service Mesh Configurations" in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide.*
- **Data Plane**: For information about Data plane configurations, see "Aspen Service Mesh Data Plane Configurations" in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide.*

4.82 Turning off AccessToken signature Validation

OAuth access tokens are used to grant an NF service consumer access to the services of an NF producer of a particular NFType for a specific period of time. With this feature, Policy has the capability to turn off AccessToken signature validation at the application layer. For instance, when service mesh, for example Aspen service mesh, is integrated with Policy, the service mesh may perform the AccessToken validation. In such cases, operator may want Policy to skip validating AccessToken signature. It checks audience and scope fields only, and send 403 - Forbidden response when any of the values do not match. In addition, when Policy receives a request without AccessToken, it sends a 401 - Unauthorized response code.

Configure

To turn of AccessToken signature validation at PCF application, the user must perform configurations that are described in the *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide.*.

4.83 Pending Transactions on N7 and N15 Interface

Policy Charging and Control (PCC) services, such as Npcf_SMPolicyControl, Npcf_AMPolicyControl service, etc. allow the NF producer to update a given policy association in the following two ways:

1. **Unsolicited** – In the unsolicited method, NF producer provides updates to the NF consumer using UpdateNotify service operation. For instance, if PCF is a NF producer that needs to notify SMF (NF consumer) about a policy association in an unsolicited manner, then PCF initiates UpdateNotify service operation for Npcf_SMPolicyControl service.

2. **Solicited** – In the solicited method, NF producer sends a response on receiving a request from the NF consumer. For instance, PCF installs policy decisions on receiving Npcf_SMPolicyControl_Update service request from SMF.

As the request can be initiated by either consumer or producer network function, a situation may arise when requests to modify the same policy are initiated concurrently. Additionally, if an HTTP proxy say Service Communication Proxy (SCP) is deployed for communication between network functions, the messages may get delivered out of order thereby resulting in wrong information maintained for a policy association by NF producer and NF consumer.

Feature Negotiation

Once the feature has been enabled, feature negotiation needs to happen between SMF and PCF during Npcf_SMPolicyControl_Create service operation. As defined by the 3GPP feature negotiation mechanism, the following conditions must be met for SMF and PCF to agree upon the Pending transactions feature:

- NF consumer or SMF advertises "PendingTransaction" feature within the attribute supportedFeatures (suppFeat) when sending a request to create SM policy association.
- In turn, PCF advertises the same value for the supportedFeatures (suppFeat) while sending the response for the policy association create request.

Enable

By default, this feature is not configured on the Policy deployment. You can opt to configure the Pending Transactions on N7 and N15 interface using the CNC Console or REST API.

On PCF Session Management and PCF Access and Mobility page, under Service Configurations, select `PendingTransaction` from the drop-down menu of **Override Supported Features** parameter. For more information about the configurations, see PCF Session Management.

Using the REST APIs for Session Management Service, you can enable the feature by updating the value as `PendingTransaction` for the following parameter under the `system` group:

```
overrideSupportedFeatures": [
      "PendingTransaction"
    ],
```

Configure

Perform the configurations for the pending transactions on N7 and N15 features using the CNC Console or REST API:

Perform the feature configurations on the PCF Session Management page. For more information about the configurations, see PCF Session Management.

Using the REST APIs for Session Management Service, you can configure the feature by updating the values of parameters under the `pendingTransaction` group:

```
pendingTransaction": {
    "updateNotifyRetryBackoff": 1000,
    "updateNotifyRetryCount": 2
  },
```

No helm configurations are required for this feature.

Observe

The existing Ingress and Egress Gateway metrics have been enhanced for this feature.

- ocpm_ingress_request_total
- ocpm_ingress_response_total
- ocpm_egress_request_total
- ocpm_egress_response_total

The following metrics have been added for this feature.

- occnp_http_in_conn_response_total
- occnp_http_in_conn_request_total
- http_bulwark_lock_request_total
- http_bulwark_unlock_request_total
- http_bulwark_lock_response_total
- http_bulwark_lock_request_retry_total
- http_bulwark_unlock_request_retry_total
- occnp_http_ue_request_total
- occnp_http_ue_response_total

No new alerts are introduced for this feature.

Maintain

Logs are generated when the system returns with a 400 error message for a pending transaction. The following is a sample log for SM update-notify request in case of Pending Transaction:

```
{
  "instant": {
    "epochSecond": 1633660911,
    "nanoOfSecond": 292427900
  },
  "thread": "boundedElastic-4",
  "level": "DEBUG",
  "loggerName": "ocpm.pcf.service.sm.domain.component.SmfManager",
  "message": "is400BadRequestWithPendingTransaction: true",
  "endOfBatch": false,
  "loggerFqcn": "org.apache.logging.slf4j.Log4jLogger",
  "threadId": 166,
  "threadPriority": 5,
  "messageTimestamp": "2021-10-07T22:41:51.292-0400"
}
```

The following is a sample log when lock is acquired for SM update request:

```
{
  "instant": {
    "epochSecond": 1633660909,
    "nanoOfSecond": 626486627
```

```
  },
  "thread": "PartitionedCacheWorker:0x0000:15",
  "level": "DEBUG",
  "loggerName": "ocpm.common.bulwark.repository.processor.LockProcessor",
  "message": "Lock is acquired for a key: bf87d498-0849-42f3-86ff-
d015a97b5721 with value: {\"lockID\":\"bb666052-270a-4e0a-
b4a5-017852c86fe3\",\"ttl\":3000,\"serviceInfo\":
{\"serviceName\":\"serviceId\",\"serviceID\":\"serviceName\"}}",
  "endOfBatch": false,
  "loggerFqcn": "org.apache.logging.slf4j.Log4jLogger",
  "threadId": 41,
  "threadPriority": 5,
  "messageTimestamp": "2021-10-08T02:41:49.626+0000"
}
```

4.84 Pending Transactions on Gx Interface

Pending transaction functionality handles race condition and retry logic for Gx messages (CCR-U and RAR).

Feature Negotiation

Once the feature has been enabled, feature negotiation needs to happen between PGW and PCRF during CCR-I operation. As defined by the 3GPP feature negotiation mechanism, the following conditions must be met for PGW and PCRF to agree upon the Pending transactions feature:

- NF consumer or PGW advertises "PendingTransaction" feature within supportedFeatures (AVP) when sending a request to create PCRF.
- In turn, PCRF advertises the same value for the supportedFeatures (AVP) while sending the response for the PGW as a CCA-I message.

Enable and Configure

By default, this feature is not configured on the Policy deployment. You can enable and configure the Pending Transactions on Gx interface using the CNC Console and REST API.

You can configure the pending transaction configurations under **Pending Transaction** section in **Settings** page for **PCRF Core** on Oracle Communications Cloud Native Core, Cloud Native Configuration Console (CNC Console). For more details, see Settings.

Using the REST APIs for Session Management Service, you can configure the feature by updating the values of parameters under the pendingTransaction group:

```
{
    "pendingTransaction": {
        "enabled": true,
        "rarRetryBackoff": "800",
        "rarRetryCount": "2",
        "userNotifyReauthErrorHandling": 1,
        "afReauthErrorHandling": 1,
        "reauthErrorHandling": 1,
        "waitGxUpdateBeforeRejectDuration": 0,
        "skipPolicyOnReauthRetry": true
    }
}
```

For more information, see "PCRF Core Service" section in Oracle Communications Cloud Native Core, Converged Policy REST API Specification Guide.

Observe

The following metrics have been added or enhanced for this feature:

- occnp_app_request_local_process_total
- occnp_app_response_local_process_total
- occnp_diam_response_local_total

For more information, see #unique_246.

4.85 NRF Client Retry and Health Check

With the alternate route retry feature, Policy can attempt service requests to an alternate secondary Network Repository Function (NRF) when the primary NRF throws errors. In addition, the health status check feature actively monitors the health of the NRFs and provides the list of the healthy NRFs for session requests only. The NRF client also provides the health information of NRFs to other services if requested, and notifies any change in the health status.

For a given service request, the NRF client initiates a request towards a healthy and the highest priority NRF. If the NRF client receives a failure response for the request or the request timed-out, it attempts to send the request to the same NRF for `NrfRetryConfig.primaryNrfRetryCount` number of times. If a success response is received before the retry count gets exhausted, NRF client accepts the response and does not send any further service requests. However, if NRF client fails to receive a success response, it attempts to send the service request to an alternate NRF. The alternate NRF is selected based on the assigned priority and health status.

If the NRF Client receives a `retryAfterTime` value in the response header from the NRF, the NRF Client halts any further attempts to the NRF and flags the NRF as unhealthy for the specified time period. The NRF client retries the service request to alternate NRFs until any one of the following conditions are met:

- NRF-client receives a success response.
- NrfRetryConfig.alternateNRFRetryCount is exhausted.
- All attempts to available healthy NRFs are exhausted.

Once any of the listed conditions are met, NRF-client accepts the response and proceeds.

NRF Client marks NRF as unhealthy under the following conditions:

- If the NRF Client receives a *retryAfterTime* value in the response header from the NRF, then NRF will be unhealthy for a time period as defined in *retryAfterTime*.
- If the status code received is available in the default values for *errorCodeReasonsForFailure*, then NRF will be unhealthy for a period of time as defined in `ConfigMap.data: profile.retryAfterTime`.
- If the status code received is available in the default values for *errorCodeReasonsForFailure* and all the retry attempts are exhausted.
- If NRF Client receives an error from Gateway service and the error is configured in the *gatewayErrorCodes* with all the exhausted retry attempts.

Note

- If NRF Client receives an error from Gateway service and the error is not configured in the *gatewayErrorCodes*, then NRF remains marked as healthy.
- *HealthCheckConfig* and *NRFRetryConfig* must be configured for the NRF Client functionality to work as expected.
- NRF Client considers a response as failure only when it is configured in the `errorReasonsForFailure` parameter in the custom-values.yaml file. The primary and non-primary NRFs must be geo-redundant for the NRF Retry mechanism to work.
- For autonomous procedures such as NfRegistration, NfHeartbeat, NfStatusSubscribe (BSF, CHF, and UDR), and NfDiscovery, NRF-client continues to retry sending service requests till a success response is received.
  For details on NRF Client configuration parameters, see *NRF Client Configuration* section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide.*

4.86 Support for PCF Status on NRF on CNC Console

Policy provides a consolidated status of the PCT registered with NRF.

The **NRF Status** is available under **Discovered NF Instances** section in **Status and Query** page on CNC Console.

Under **Discovered NF Instances** page, you can also view the health status of primary and secondary NRFs, or alternate NRF.

This feature is available for Converged and PCF only deployment modes.

To view the status using CNC Console, see NF Status.

4.87 Support for Stale Binding Detection in BSF

To support BSF to query PCF for suspected stale binding records, the Binding service has been made mandatory in the PCF mode of deployment. In BSF, whenever a binding record is detected as stale on the expiry of its binding age, BSF may send a query to PCF to confirm its status. To support this query to PCF, PCF is first required to send the **Vendor-Specific-Attribute** containing PCF Notification URL in the Binding Register request. When PCF sends the **Vendor-Specific-Attribute** in the register request, it is also required to provide the vendorID.

To configure this support, the following two new parameters are added to the Binding Service configurations page:

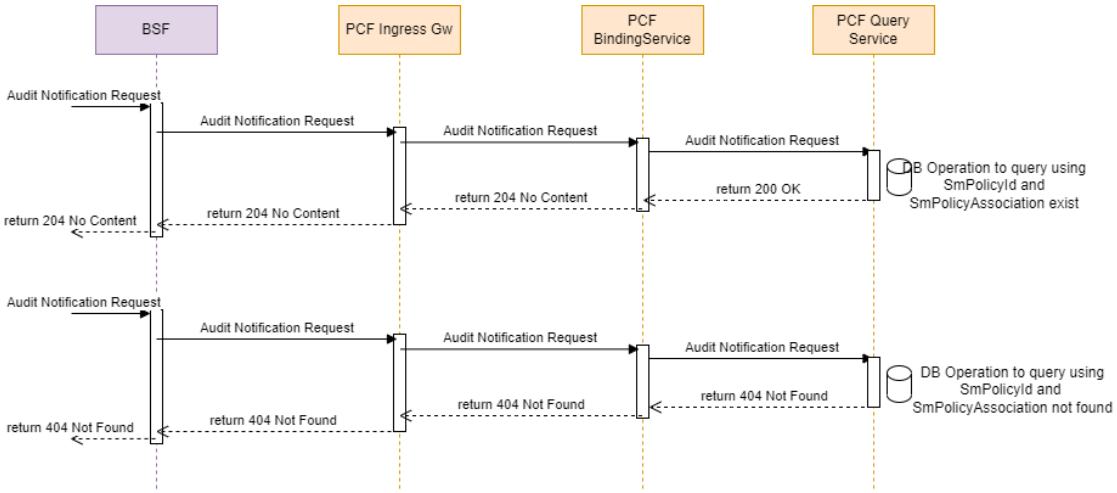
- Enable Vendor-Specific-Attribute In Register Request
- VendorId

Considering that the support for stale binding detection in BSF is enabled, the following diagram describes the call flow between BSF ad PCF. BSF on receiving an Audit notification request, sends the Audit Notification request to PCF Binding Service via Ingress Gateway. Then, Binding Service sends a request to the PCF Query Service to check if the ContextId (SmPolicyAssociation) exists. If it exists, PCF Bindig service responds with 200 OK.

If the ContextId (SmPolicyAssociation) does not exist, Query service sends 404 Not Found to Binding Service. Then, Binding service returns the same response to BSF.

In case Policy is unable to process the Audit Notification request due to any failures, then it returns the 500 Internal Server error response to BSF.

Figure 4-105 BSF and PCF Interaction for Stale Binding Detection



Based on the system configurations, Policy supports receiving `3gpp-Sbi-Binding` header in the Audit Notification requests from BSF. Then, it updates the record in its database and send the updated `3gpp-Sbi-Binding` header in the response for Audit Notification request.

Metrics

The following metrics are implemented for this feature:

- occnp_binding_query_request_count
- occnp_binding_query_response_count

Logs

Policy generates logs for request and response messages from BSF including the VSA attribute. In addition, it generates logs for audit notifications requests and responses sent for it.

4.88 Pod Protection at Ingress Gateway

This section describes how to protect the Ingress Gateway pods when they are overloaded with numerous incoming requests.

The Ingress Gateway pods are not protected against any incoming traffic congestion. As a result, the pods are overloaded and congested. This impacts system latency and performance. It also leads to stability issues due to uneven distribution of connections and traffic on Ingress Gateway pods. As a front end microservice for HTTP traffic, it is important for Ingress Gateway to have pod protection implemented.

To configure pod protection on Ingress Gateway, you can define threshold limit for DoC and Congested state through REST:

Level	Resource
DoC	CPU
	Memory
	Pending Message
Congested	CPU
	Memory
	Pending Message

Configure

You need to perform the following configurations for pod protection feature:

- **Configure using REST API:** Policy provides the following REST API: {apiRoot}/PCF/nf-common-component/v1/igw/podprotection

  You can perform the GET, PATCH, or PUT operation to configure the feature. For more information about REST API configuration, see *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

4.89 Support for Concurrency Handling using Bulwark Service in Policy

Policy supports the Bulwark service to handle the concurrent requests coming from other Policy services. Bulwark Service provides lock and unlock mechanism over a SUPI or GPSI key and allows only one notification at a time to proceed.

The Policy Services (SM and PDS service) sends the following parameters to request a lock:

- **Key:** The key is the identifier value with which the lock is identified. You can select either SUPI or GPSI value to be the key parameter for which the lock acquisition request must be made. By default, SUPI values are used for lock requests.
- **Lease Duration:** This value defines the duration for which lock is kept once the acquisition is successful. After this duration, the lock will be released automatically.
- **Lock Wait Timeout:** This defines the duration by which the Policy services wait for the response to get a lock. The same duration is used by Bulwark to poll for the lock in case the lock is not available. The Lock Wait timeout is also considered as the polling interval which sends the request towards Bulwark service periodically to acquire the lock for another create or delete request.
- **Retry Count:** This defines the count of retry attempts that are made by the Policy services if a lock request fails. In case all the retry attempts fail, the create or delete request is rejected. By default, 3 retry attempts are enabled.

Lease Duration Range at Bulwark
In order to guard bulwark service in terms of lease duration values, a minimum and a maximum lease duration value has been added. Below are the default values is in milliseconds.

```
bulwark:
  leaseDurationMs:
    min: 10
    max: 15000
```

If the incoming request lease duration value is higher than the maximum configured lease duration in Helm, then the lease duration is overwritten with maximum configured value. And, if the incoming request lease duration value is lower than the the minimum configured lease duration in Helm, then the lease-duration is overwritten with minimum configured value.

> **Note**
>
> Since these are engineering driven values for safeguarding the bulwark service it is not exposed in the custom.yaml file. Please contact Policy engineering team in case any modification is required.

4.89.1 Support for Concurrency Handling using Bulwark Service in SM

The Bulwark Service integrates with the SM service to handle the concurrent requests for the SM Create, SM Update, SM Delete, SM Update-Notify, and SM Clean procedures. The SM service integration with Bulwark service handles the concurrent requests for the same subscriber in a concurrent and efficient manner.

In case of multiple create, update, delete, or clean requests for the same subscriber, the SM service requests a lock for the SUPI or GPSI value from the Bulwark service. Once the lock is acquired, the SM create, SM Update, SM delete, SM Update-Notify, or SM Clean request is executed for the SUPI or GPSI value with which the lock is requested. After the successful completion of the requests, the SM service sends a request to release the lock to Bulwark.

As the notification is locked, pending transaction comes into picture.

When Bulwark-Service is Enabled, the following scenarios can occur between pending transaction and SM Update:

Table 4-44 Pending Transaction and Concurrency in SM Update

Pending Transaction	Concurrency in SM Update	Use-Case
Enabled	Disabled	Single Key - Session level lock will be acquired. Pending-transaction call flow would work.
Disabled	Enabled	Single Key - Subscriber level lock will be acquired. Update-Notify call flow with concurrency would work.
Enabled	Enabled	Multi-Key - Session Level & Subscriber Level locks will be acquired.

When Bulwark-Service is Enabled, the following scenarios can occur between pending transaction and update-notify:

Table 4-45 Pending Transaction and Concurrency in SM Update-Notify

Pending Transaction	Concurrency in Update-Notify	Use-Case
Enabled	Disabled	Single Key - Session level lock will be acquired. Pending-transaction call flow would work.

Table 4-45 (Cont.) Pending Transaction and Concurrency in SM Update-Notify

Pending Transaction	Concurrency in Update-Notify	Use-Case
Disabled	Enabled	Single Key - Subscriber level lock will be acquired. SM-Update call flow would work.
Enabled	Enabled	Multi-Key - Session Level & Subscriber Level locks will be acquired.

When Bulwark-Service is Enabled, the following scenarios can occur between pending transaction and SM Update and SM update-notify:

Table 4-46 Pending Transaction and Concurrency in SM Update as well as SM Update-Notify

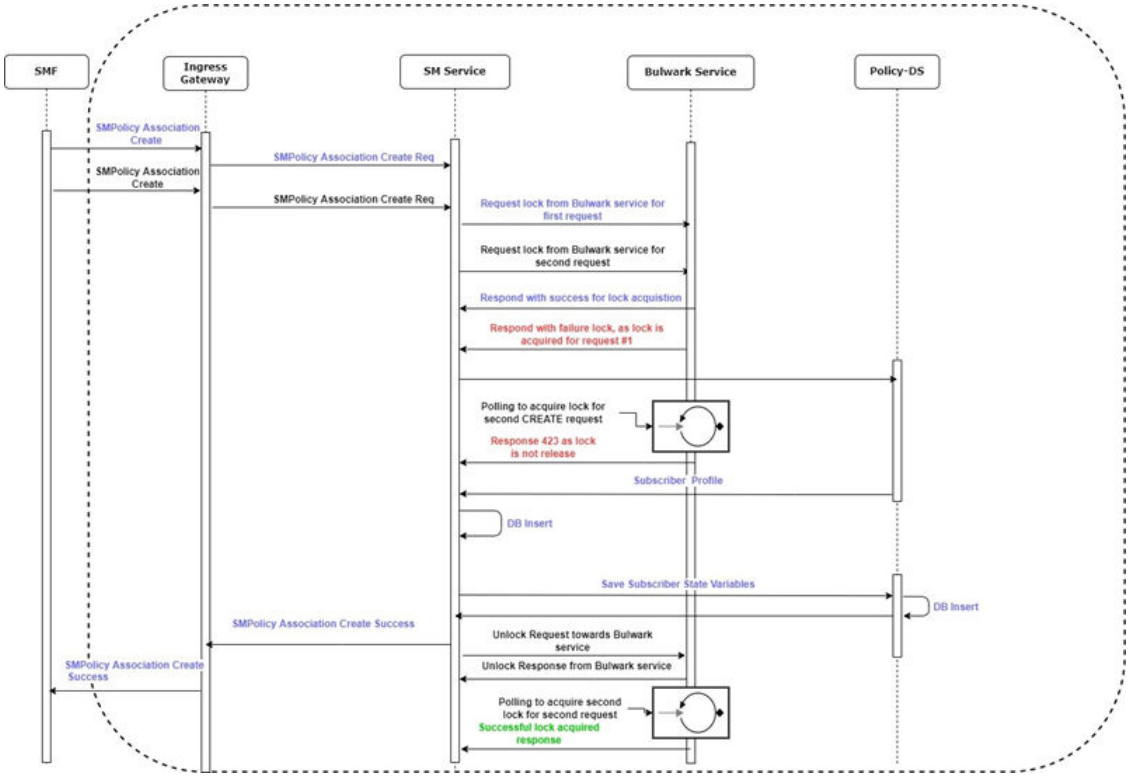
Pending Transaction	Concurrency in SM Update	Concurrency in SM Update-Notify	Use-Case
Enabled	Disabled	Disabled	Session Level Lock Acquired - during collision, lock acquisition fails at SessionID
Disabled	Enabled	Disabled	Subscriber Level lock acquired - during collision, lock acquisition fails at subscriberID
Disabled	Enabled	Enabled	Subscriber Level lock acquired - during collision, lock acquisition fails at subscriberID
Enabled	Enabled	Disabled	Session Level & Subscriber Level locks will be acquired - during collision, lock acquisition will fail at SessionID

Call Flow for handling concurrent requests for SM Service

Call Flow for SM Create Request

The following diagram shows a scenario for SM create call flow where no service responds with error:

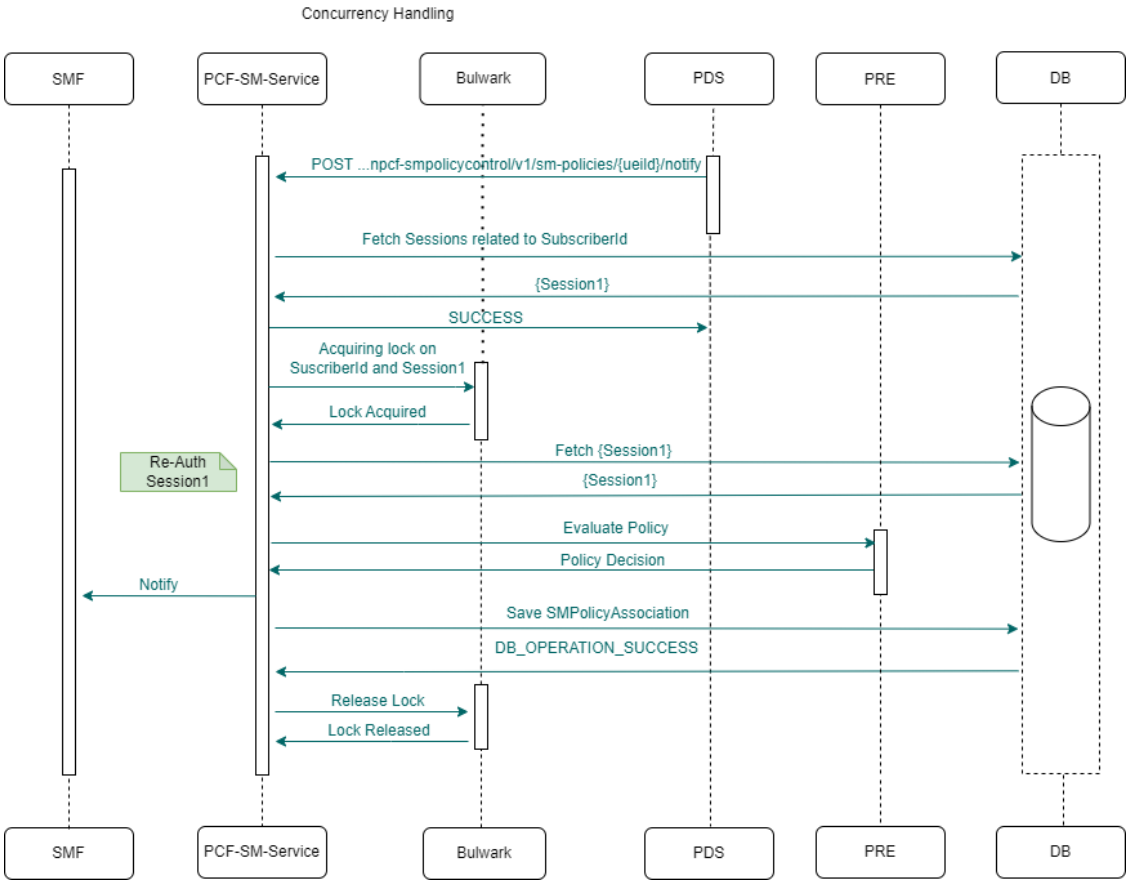
Figure 4-106 Call Flow - Bulwark Lock Request SM Create Procedure



Call Flow for SM Update-Notify Request

The following diagram shows a scenario for SM Update-Notify call flow for one notification and one session:

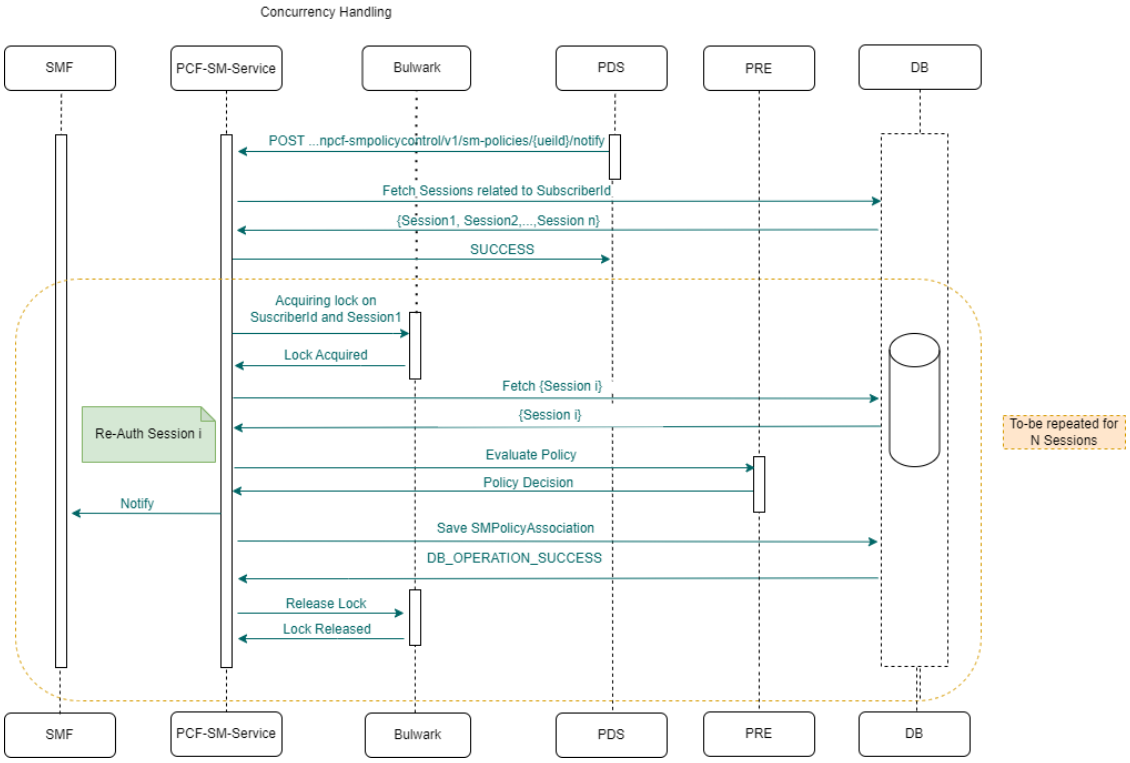
Figure 4-107 Call Flow for SM Update-Notify Request (One Notification and One Session)



1. PDS sends a notify request to SM service.
2. SM service fetches the corresponding sessions from database and sends a success response to PDS.
3. A notify request tries to acquire the lock from bulwark service and gets the lock.
4. The notify request starts processing by fetching the session.
5. SM service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
6. SM policy association is saved in DB.
7. On successful save, a notify response is sent to SMF and the lock is released.

The following diagram shows a scenario for SM Update-Notify call flow for one notification and n sessions:

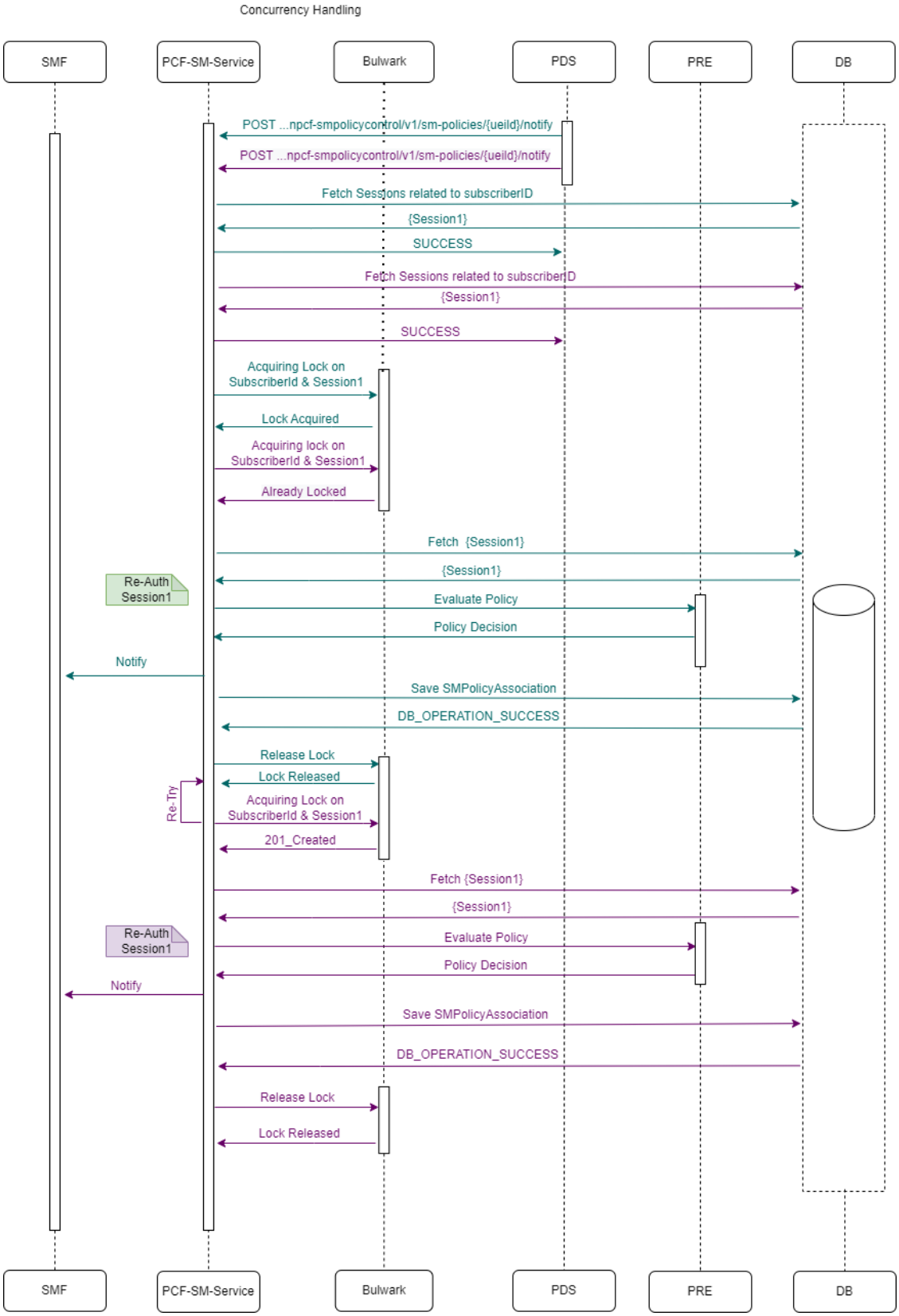
Figure 4-108 Call Flow for SM Update-Notify Request (One Notification and n Sessions)



1. PDS sends a notify request to SM service.
2. SM service fetches the corresponding sessions from database and sends a success response to PDS.
3. A notify request tries to acquire the lock from bulwark service and gets the lock.
4. The notify request starts processing by fetching each sessions one by one.
5. SM service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
6. SM policy association is saved in DB.
7. On successful save, a notify response is sent to SMF and the lock is released.
8. This process gets repeated for n sessions of the Notify request.

The following diagram shows a scenario for SM Update-Notify call flow for two notifications and one session:

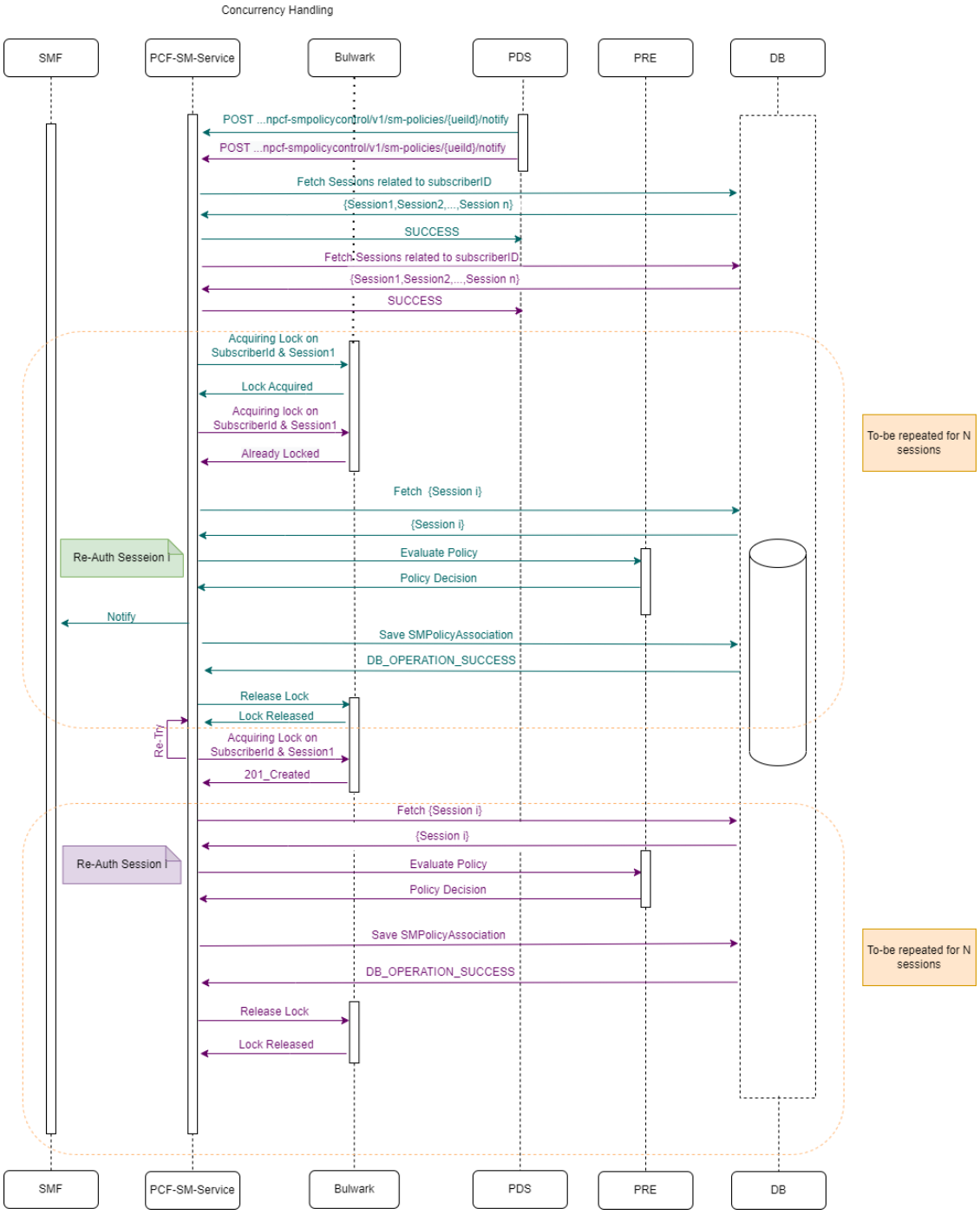
Figure 4-109 Call Flow for SM Update-Notify Request (Two Notifications and 1 Session)



1. PDS sends a notify request to SM service.
2. SM service fetches the corresponding sessions from database and sends a success response to PDS.
3. First notify request tries to acquire the lock from bulwark service and gets the lock.
4. Meanwhile, if second notify request tries to acquire lock, it will get an already locked response from Bulwark.
5. First notify request starts processing by fetching each sessions one by one.
6. SM service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
7. SM policy association is saved in DB.
8. On successful save, a notify response is sent to SMF and the lock is released.
9. Then the second notify request will retry to acquire the lock. On lock acquisition, the same process is followed for second request as well.

The following diagram shows a scenario for SM Update-Notify call flow for two notifications from same subscriber and n sessions each:

Figure 4-110 Call Flow for SM Update-Notify Request (Two Notifications from Same Subsriber and n Sessions Each)

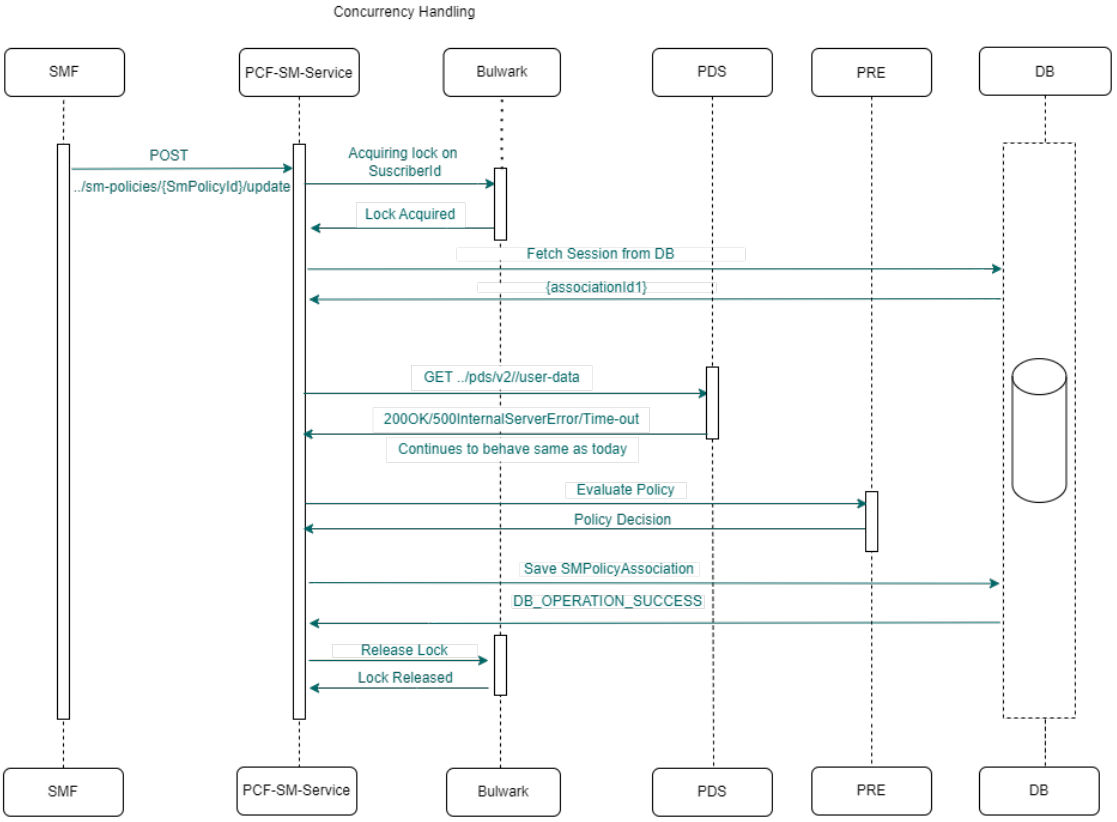


1. PDS sends a notify request to SM service.
2. SM service fetches the corresponding sessions from database and sends a success response to PDS.
3. First notify request tries to acquire the lock from bulwark service and gets the lock.

4. Meanwhile, if second notify request tries to acquire lock, it will get an already locked response from Bulwark.
5. First notify request starts processing by fetching each sessions one by one.
6. SM service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
7. SM policy association is saved in DB.
8. On successful save, a notify response is sent to SMF and the lock is released.
9. This process gets repeated for n sessions of first Notify request.
10. Then the second notify request will retry to acquire the lock. On lock acquisition, the same process is followed for second request as well.

Call Flow for SM Update Request

Figure 4-111 Concurrency enabled for SM Update for a single session

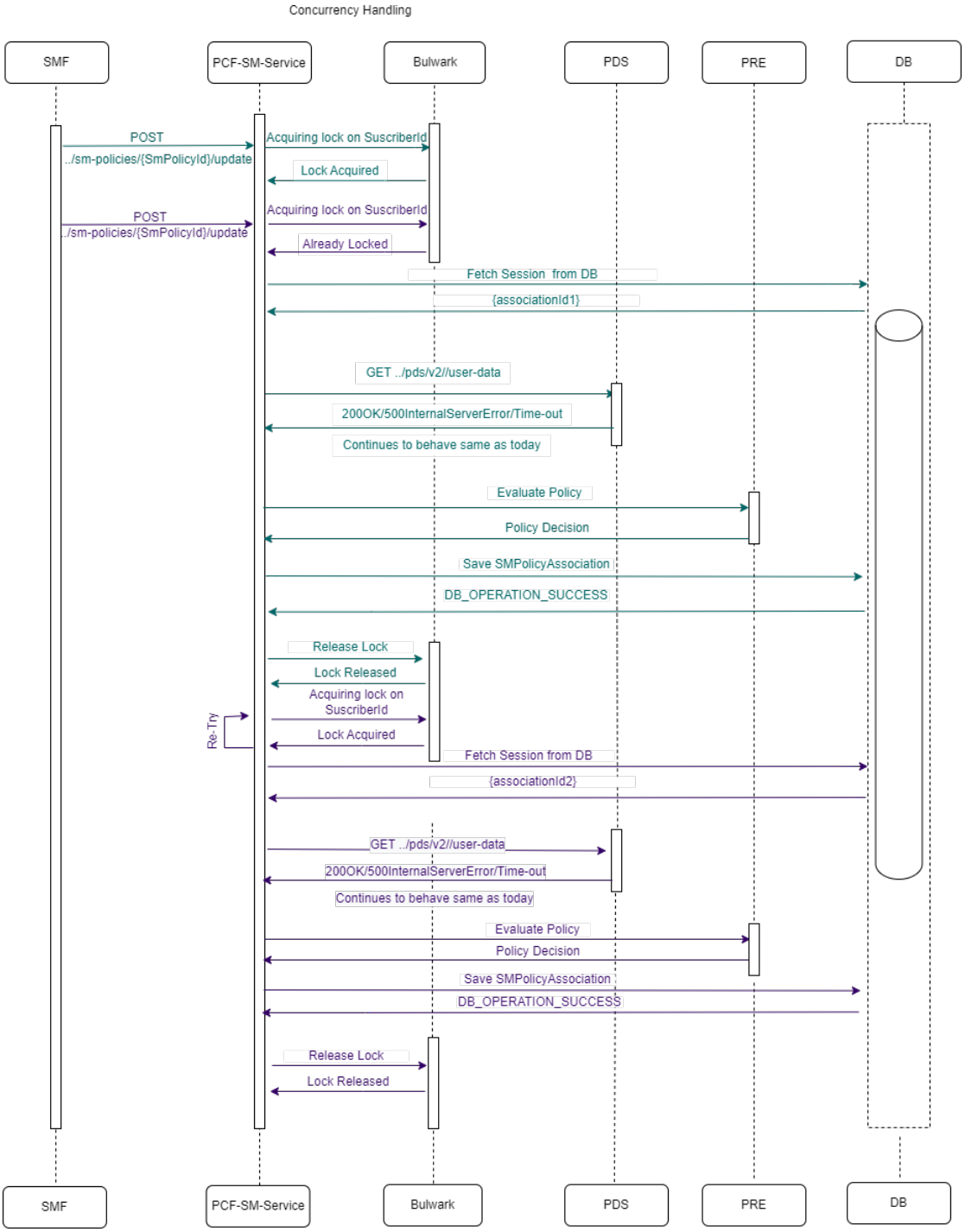


In this case, only concurrency is enabled for SM update and pending transaction is disabled. SM Service acquires a single key lock using subscriber ID of the request.

1. SM Service receives an SM Update request from SMF.
2. SM Service sends a lock request to Bulwark Service based on Subscriber ID.
3. SM Service acquires a lock from Bulwark Service.
4. SM Service fetches the session details from the database and sends the update request to PDS.

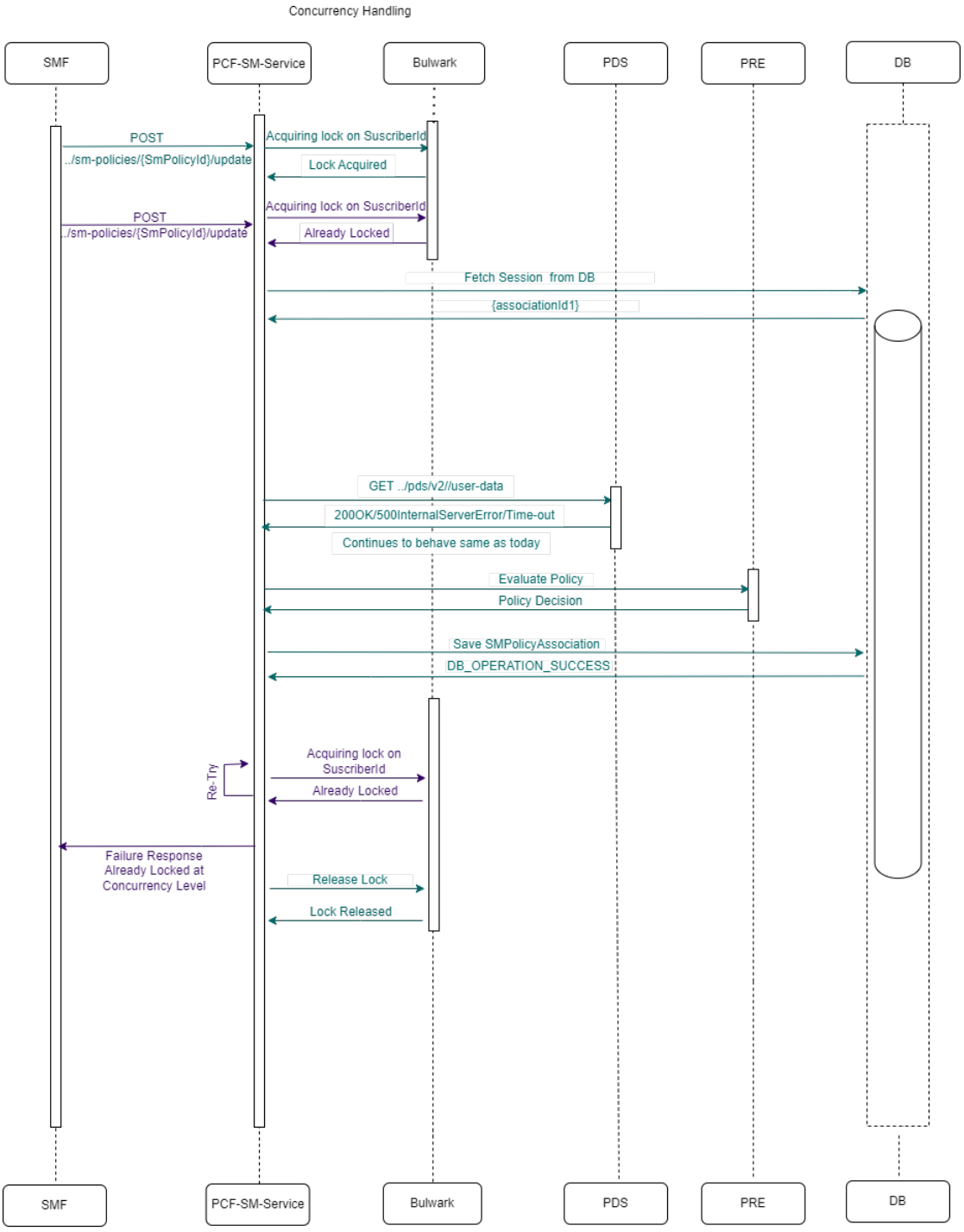
5. SM service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
6. SM policy association is saved in the database.
7. After the successful completion of the updates, SM Service sends a lock release request to Bulwark Service to release the lock.

Figure 4-112 Concurrency Enabled for SM-Update and multiple updates for the same subscriber



1. SM Service receives the first SM Update request from SMF.
2. SM Service sends a lock request to Bulwark Service based on Subscriber ID.
3. SM Service acquires a lock for first request from Bulwark Service.
4. SM Service receives second SM Update request from SMF for the same subscriber.
5. SM Service sends a lock request to Bulwark Service for the second SM Update request.
6. Bulwark Service responds to SM Service for the second request with a 423 already locked message.
7. SM Service fetches the session details from the database for the first update request and sends the update request to PDS.
8. SM service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
9. SM policy association is saved in the database.
10. After the successful completion of the updates, SM Service sends a lock release request to Bulwark Service to release the lock for the first update request.
11. SM Service retries to acquire the lock for the second update request from Bulwark Service.
12. SM Service acquires a lock for the second update request, processes the update request.
13. After completing the updates for the second request, SM Service sends an unlock request to Bulwark Service to release the lock for the second request.

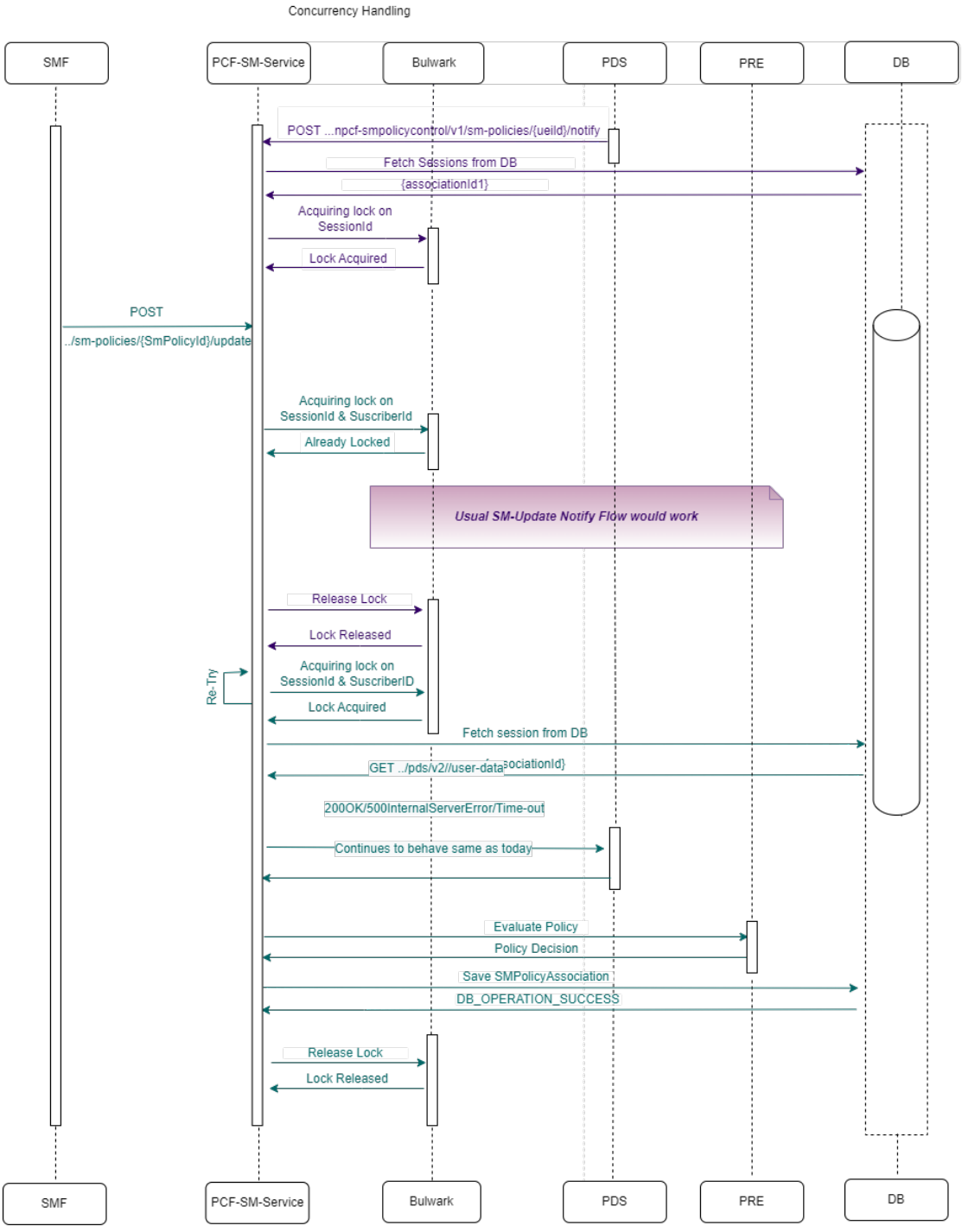
Figure 4-113 Retrial failure when there are multiple updates for same subscriber in different sessions



1. SM Service receives the first SM Update request from SMF.
2. SM Service sends a lock request to Bulwark Service based on Subscriber ID.
3. SM Service acquires a lock for first request from Bulwark Service.
4. SM Service receives second SM Update request from SMF for the same subscriber.

5. SM Service sends a lock request to Bulwark Service for the second SM Update request.
6. Bulwark Service responds to SM Service for the second request with a 423 already locked message.
7. SM Service fetches the session details from the database for the first update request and sends the update request to PDS.
8. SM service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
9. SM policy association is saved in the database.
10. SM Service retries to acquire the lock for the second update request from Bulwark Service.
11. SM Service receives a 423 already acquired lock for the second lock request from Bulwark Service.
12. After the successful completion of the updates for the first request, SM Service sends a lock release request to Bulwark Service to release the lock for the first update request.

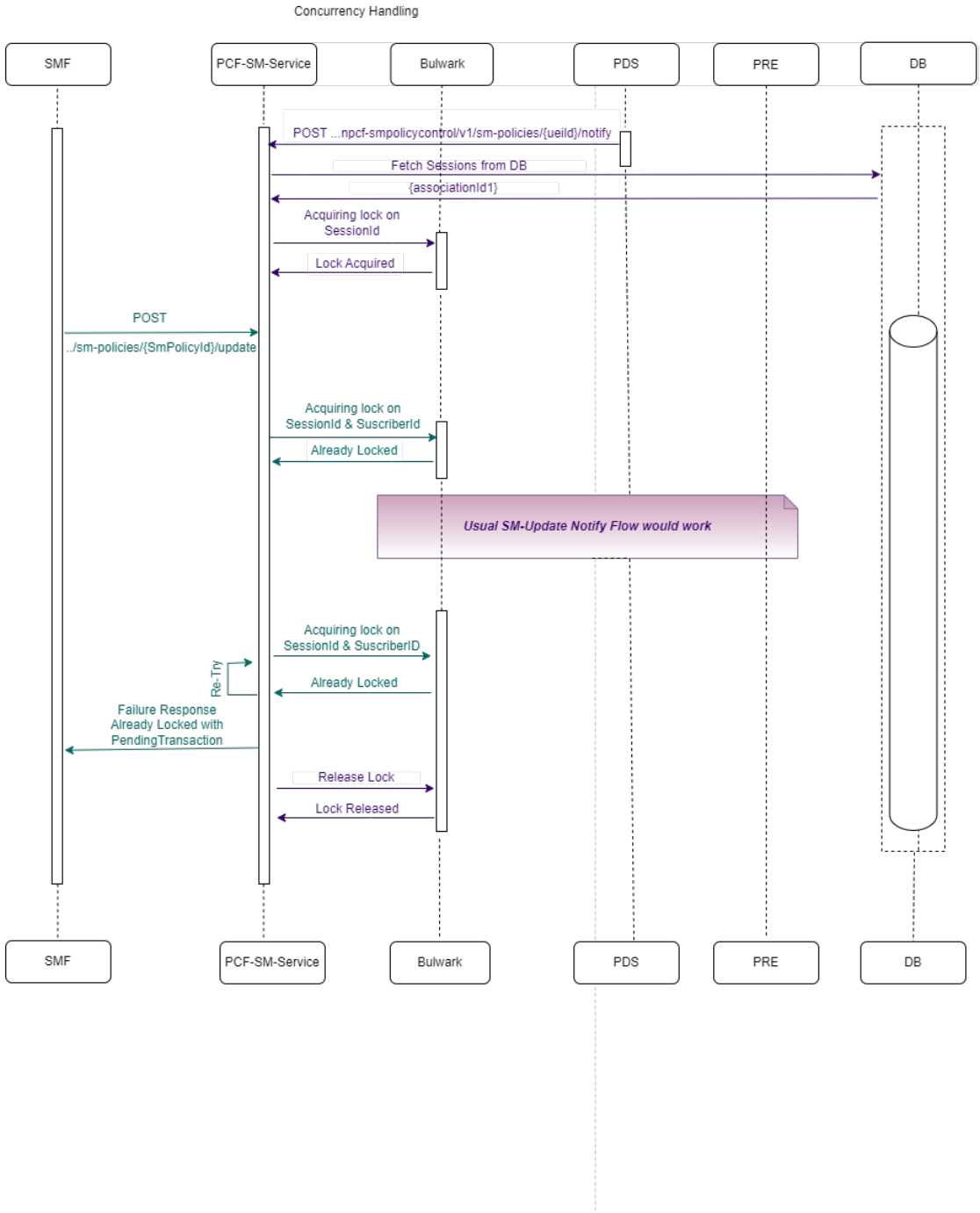
Figure 4-114 Pending transaction and Concurrent SM-Update for same session - Success scenario



1. PDS sends a notify request to SM service.
2. SM service fetches the corresponding sessions from database and sends a success response to PDS.
3. SM Service tries to acquire a lock from Bulwark Service using SessionID and gets the lock.

4. SM Service receives an SM Update request from SMF.
5. SM Service sends a lock request to Bulwark Service using the Session ID as well as Subscriber ID.
6. Bulwark Service responds to SM Service for the second request with a 423 already locked message.
7. After the successful completion of the first SM Update-notify request, SM Service sends an unlock request to Bulwark Service and Bulwark Service releases the lock for the first SM Update-Notify request.
8. SM Service retries to acquire the lock for the SM-Update request from SMF using the same Session ID as well as Subscriber ID.
9. SM Service acquires a lock with the retry.
10. SM Service fetches the session details from the database for the first update request and sends the update request to PDS.
11. SM service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
12. SM policy association is saved in the database.
13. After the successful completion of the updates for the SM Update request, SM Service sends a lock release request to Bulwark Service to release the lock.

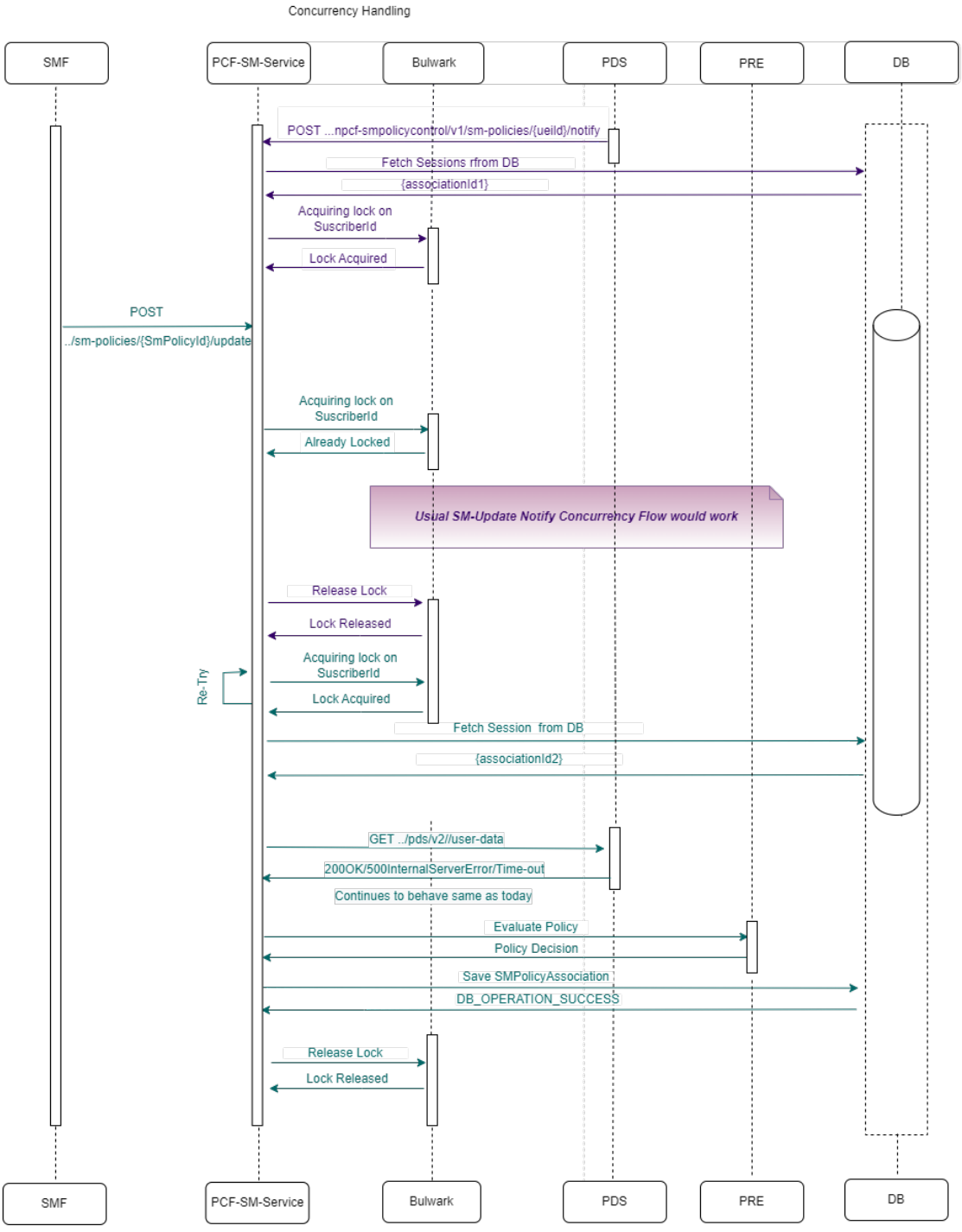
Figure 4-115 Pending transaction and Concurrent SM-Update for same session - Retry failure scenario



1. PDS sends a notify request to SM service.
2. SM service fetches the corresponding sessions from database and sends a success response to PDS.
3. SM Service tries to acquire a lock from Bulwark Service using SessionID and gets the lock.
4. SM Service receives an SM Update request from SMF.

5. SM Service sends a lock request to Bulwark Service using the Session ID as well as Subscriber ID.
6. Bulwark Service responds to SM Service for the second request with a 423 already locked message.
7. SM Service processes the SM Update-notify request.
8. SM Service retries to acquire the lock for the SM-Update request from SMF using the same Session ID as well as Subscriber ID.
9. Bulwark Service responds to SM Service for the second request with a 423 already locked message.
10. If the retry count exhausts, SM Service sends an already locked with pending transaction - failure response to SMF.
11. After the successful completion of the updates for the SM Update-Notify request, SM Service sends a lock release request to Bulwark Service to release the lock.

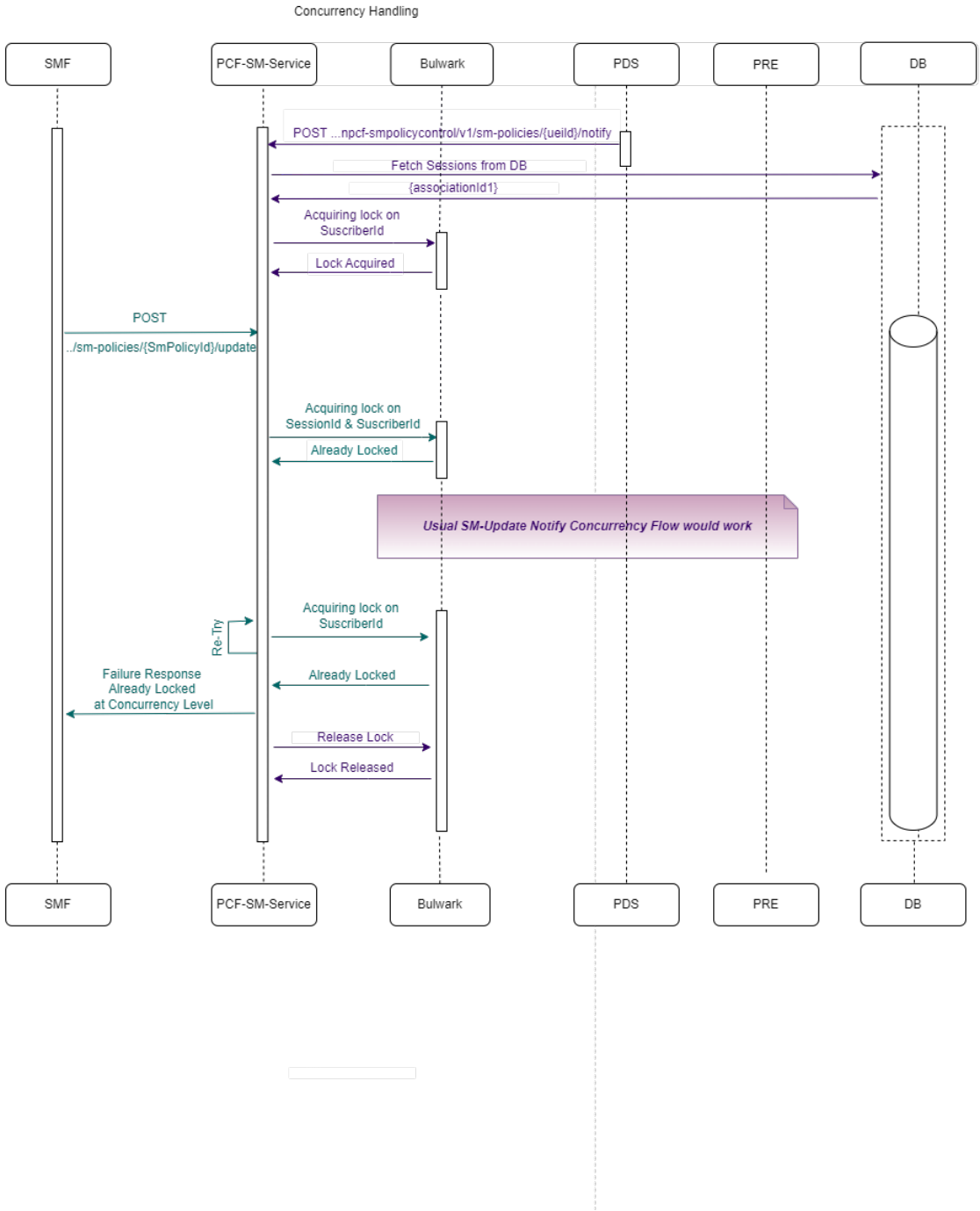
Figure 4-116 Concurrent SM-Update-Notify and Concurrent SM-Update - Success scenario



1. PDS sends a notify request to SM service.
2. SM service fetches the corresponding sessions from database and sends a success response to PDS.
3. SM Service tries to acquire a lock from Bulwark Service using SubscriberID and gets the lock.

4. SM Service receives an SM Update request from SMF.
5. SM Service sends a lock request to Bulwark Service for the same Subscriber ID.
6. Bulwark Service responds to SM Service for the second request with a 423 already locked message.
7. SM Service processes the SM Update-notify request.
8. After the successful completion of the updates for the SM Update-Notify request, SM Service sends a lock release request to Bulwark Service to release the lock.
9. SM Service retries to acquire the lock for the SM-Update request from SMF using the same Subscriber ID and acquires the lock.
10. SM Service fetches the session details from the database for the second update request and sends the update request to PDS.
11. SM service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
12. SM policy association is saved in the database.
13. After the successful completion of the updates for the second request, SM Service sends a lock release request to Bulwark Service to release the lock.

Figure 4-117 Concurrent SM-Update-Notify and Concurrent SM-Update - Failure scenario



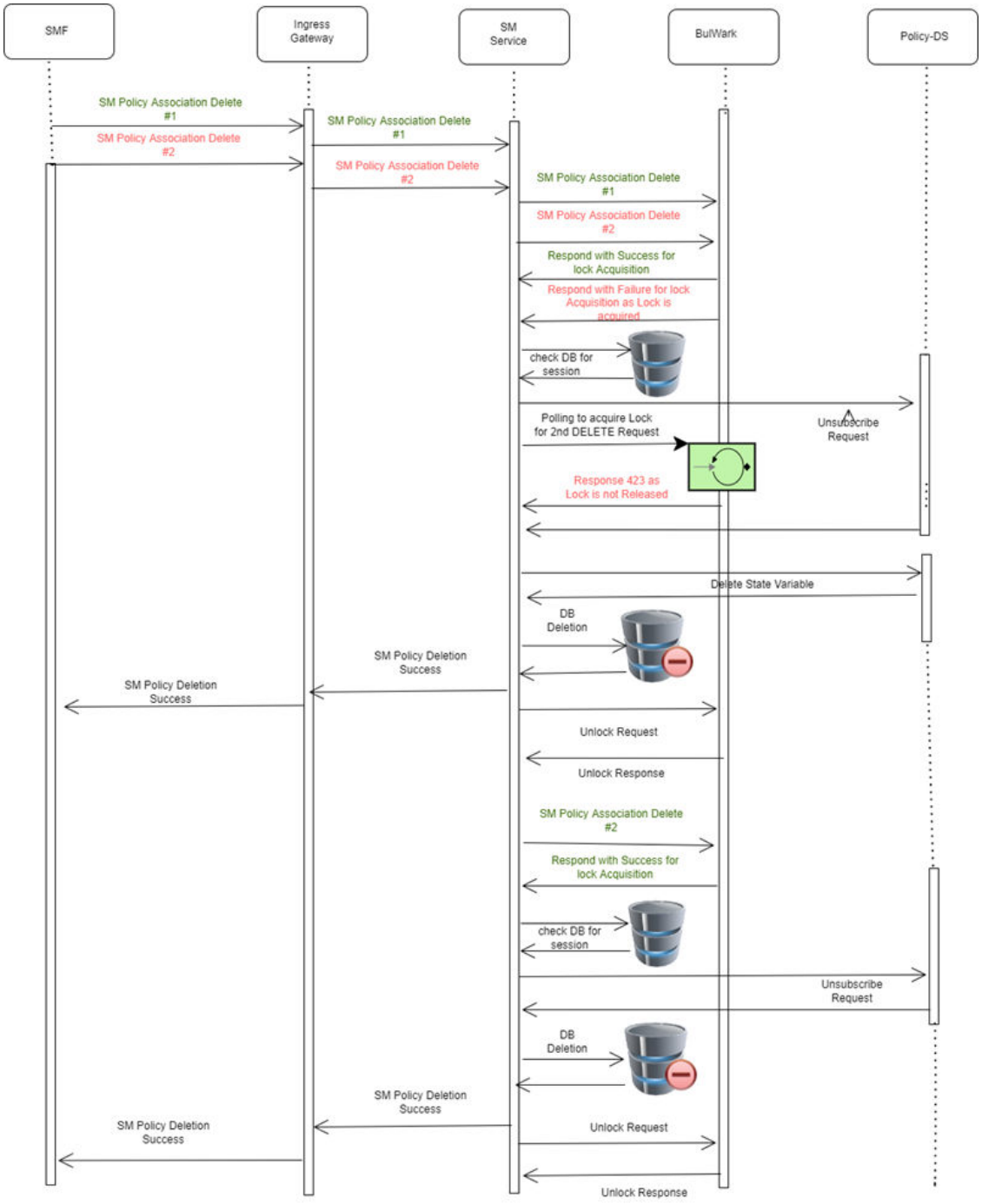
1. PDS sends a notify request to SM service.
2. SM service fetches the corresponding sessions from database and sends a success response to PDS.
3. SM Service tries to acquire a lock from Bulwark Service using SubscriberID and gets the lock.

4. SM Service receives an SM Update request from SMF.
5. SM Service sends a lock request to Bulwark Service for the same Subscriber ID.
6. Bulwark Service responds to SM Service for the second request with a 423 already locked message.
7. SM Service processes the SM Update-notify request.
8. SM Service retries to acquire the lock for the SM-Update request from SMF using the same Subscriber ID.
9. Bulwark Service responds to SM Service for the second request with a 423 already locked message.
10. After the successful completion of the updates for the SM Update-Notify request, SM Service sends a lock release request to Bulwark Service to release the lock.

Call Flow for SM Delete Request

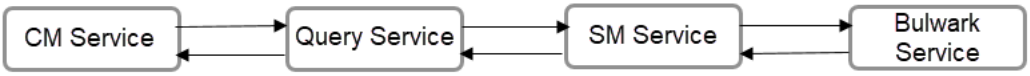
The following diagram shows a scenario for SM delete call flow where no service responds with error:

Figure 4-118 Call Flow - Bulwark Lock Request SM Delete Procedure



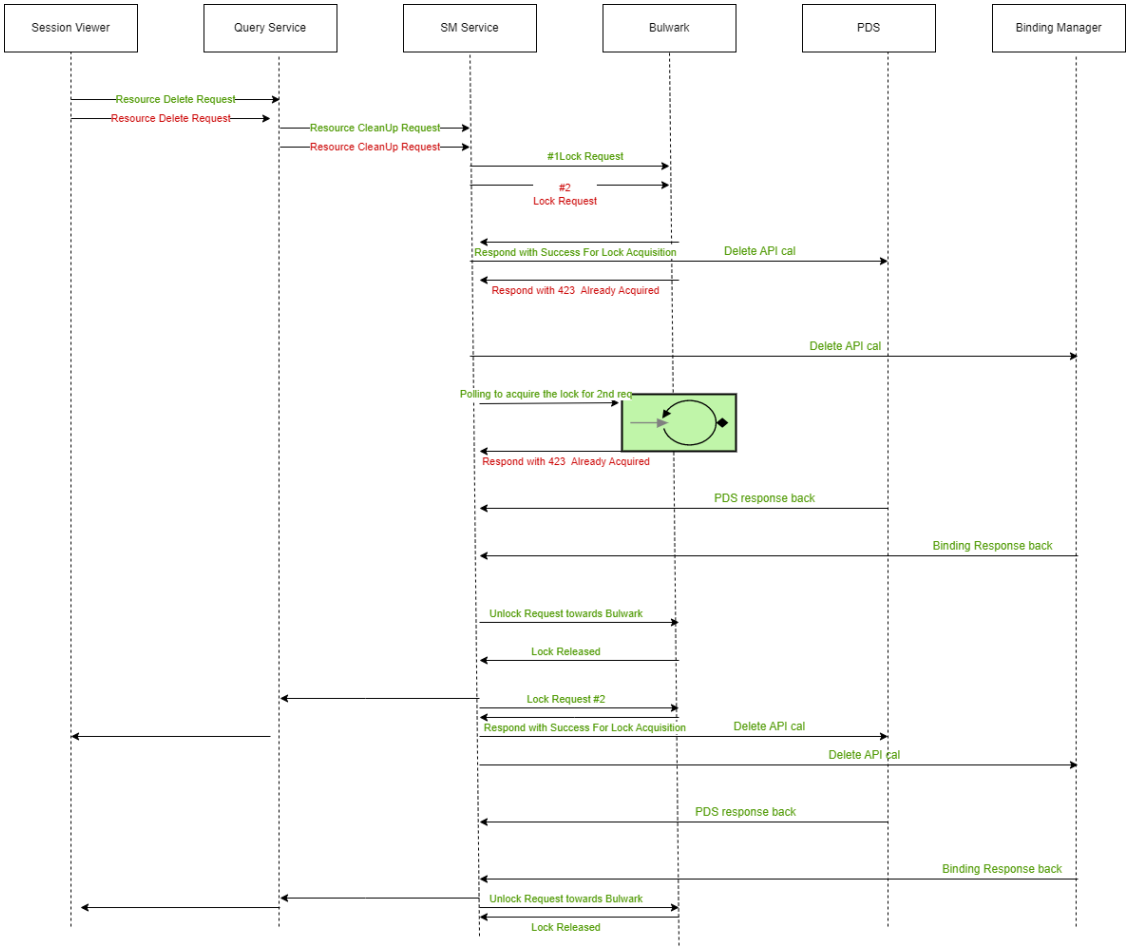
Call Flow for SM Cleanup (Session Viewer Delete)

Figure 4-119 Requests for SM Clean



The cleanup can be either a local delete or a remote delete. In either case, SM Cleanup API is called to delete the resource.

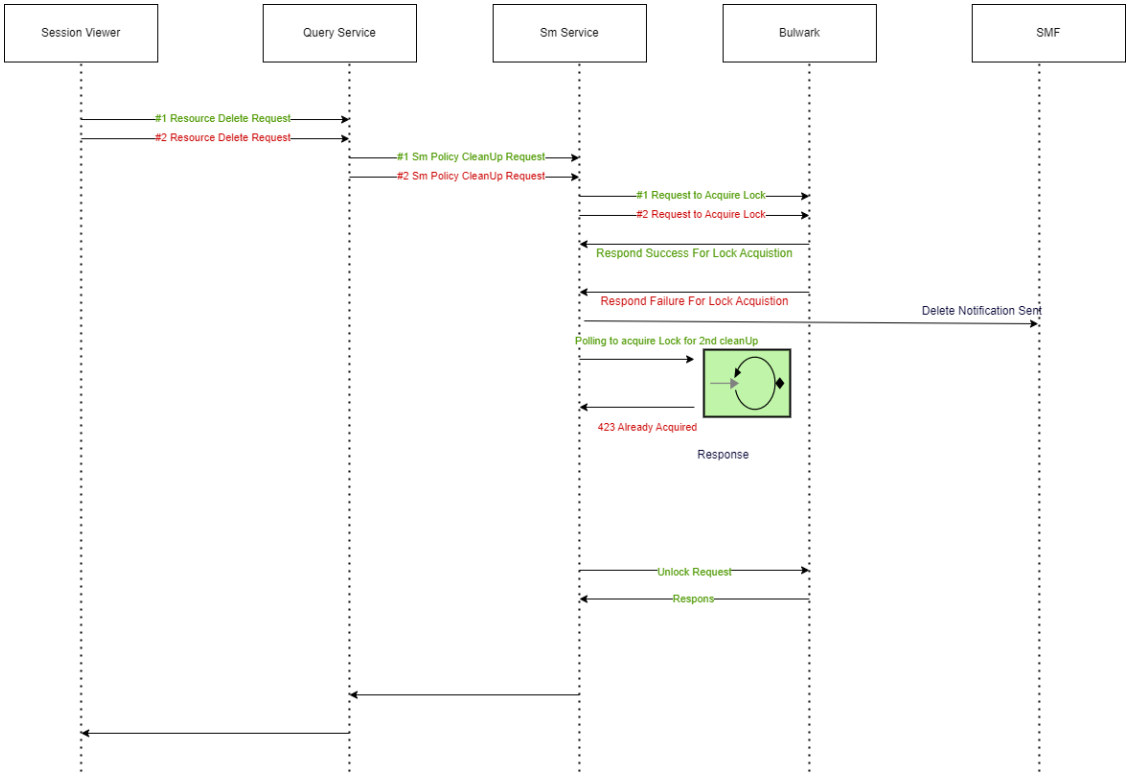
Figure 4-120 Local delete



1. CM Service (**Session Viewer**) sends a Resource Delete Request based on policy association ID to Query Service, which in turn sends the Resource Cleanup request to SM Service.
2. If `CONCURRENCY.BULWARK_ENABLED_FOR_N7_CLEANUP` is enabled, SM Service sends a lock request to Bulwark Service to acquire the lock for the cleanup request.
   SM Service extracts the subscriber value using the policy association ID and uses the same as the key to gain the lock from Bulwark Service.
3. If the lock acquisition is successful:
   a. SM Service sends a delete request to PDS Resource Delete API to delete SM service's user data related to the individual SM policy association or the subscriber.
   b. After the data deletion, PDS responds to SM Service.
   c. SM service sends a delete request to Binding Service Resource Delete API to delete the binding data related to the individual SM policy association or the subscriber.
   d. After the data deletion, Binding Service responds to SM Service.
4. After receiving response from PDS and Binding Service, SM Service sends an unlock request to Bulwark Service to release the lock.

5. If SM Service receives another cleanup request at the same time, it tries to gain a lock for this second request from Bulwark Service.
6. If the lock request fails, SM Service receives a 423 already acquired response from Bulwark Service.
7. As per the configured number of retry counts, SM Service retries to gain the lock.
8. If it gains a lock, the SM Service sends a delete request to PDS and to Binding Service to delete the data.
9. If it fails to gain a lock and the number of retries reaches `CONCURRENCY.LOCK_REQUEST_RETRY_COUNT_FOR_CLEANUP`, Bulwark Service sends a 500 internal error to SM Service.

Figure 4-121 Remote delete



SM Service cleans up the data remotely when you check **Remote** checkbox in **Session Viewer** page on the UI and then click **Delete**.

1. CM Service (**Session Viewer**) sends a Resource Delete Request based on policy association ID and `deleteScope=Remote` to Query Service, which in turn sends the Resource Cleanup request to SM Service.
2. If `CONCURRENCY.BULWARK_ENABLED_FOR_N7_CLEANUP` is enabled, SM Service sends a lock request to Bulwark Service to acquire the lock for the cleanup request.
   SM Service extracts the subscriber value using the policy association ID and uses the same as the key to gain the lock from Bulwark Service.
3. If the lock acquisition is successful:
   a. SM Service sends a Session Delete Notification to SMF.

   b. If SM Service, receives a `404 not found` response, it emulates a Session Delete Request to delete the SM Policy Association.
   c. SM Service deletes the SM Policy Association as per the PA Session Delete Procedure.
4. After the deletion is successful, SM Service sends an unlock request to Bulwark Service to release the lock.
5. If SM Service receives another cleanup request at the same time, it tries to gain a lock for this second request from Bulwark Service.
6. If the lock request fails, SM Service receives a 423 already acquired response from Bulwark Service.
7. As per the configured number of retry counts, SM Service retries to gain the lock.
8. If it gains a lock, the SM Service sends a Session Delete Notification to SMF and deletes the SM Policy Association as per the PA Session Delete Procedure.
9. If it fails to gain a lock and the number of retries reaches `CONCURRENCY.LOCK_REQUEST_RETRY_COUNT_FOR_CLEANUP`, Bulwark Service sends a 500 internal error to SM Service.

Managing Concurrency Handling for SM Service

Enable

You can enable the concurrency handling through Bulwark using the CNC Console or REST API for Policy.

- **Enable using CNC Console:** Set value for the the following parameters under the **Advanced Settings** section on the **PCF Session Management** page:
  - CONCURRENCY.BULWARK_ENABLED_FOR_N7_CREATE
  - CONCURRENCY.BULWARK_ENABLED_FOR_N7_DELETE
  - CONCURRENCY.LOCK_LEASE_DURATION_FOR_UPDATE_NOTIFY
  - CONCURRENCY.LOCK_WAIT_DURATION_FOR_UPDATE_NOTIFY
  - CONCURRENCY.LOCK_REQUEST_RETRY_COUNT_FOR_UPDATE_NOTIFY
  - CONCURRENCY.BULWARK_ENABLED_FOR_N7_UPDATE_NOTIFY
  - CONCURRENCY.UPDATE_USERDATA_IN_ASSOCIATION_ON_LOCKFAILURE
  - CONCURRENCY.LOCK_LEASE_DURATION_FOR_UPDATE
  - CONCURRENCY.LOCK_WAIT_DURATION_FOR_UPDATE
  - CONCURRENCY.LOCK_REQUEST_RETRY_COUNT_FOR_UPDATE
  - CONCURRENCY.BULWARK_ENABLED_FOR_N7_UPDATE
  - CONCURRENCY.LOCK_LEASE_DURATION_FOR_CLEANUP
  - CONCURRENCY.LOCK_WAIT_DURATION_FOR_CLEANUP
  - CONCURRENCY.LOCK_REQUEST_RETRY_COUNT_FOR_CLEANUP
  - CONCURRENCY.BULWARK_ENABLED_FOR_N7_CLEANUP

> **Note**
>
> If you do not make these parameters as configurable, the SM service uses the default values for these parameters.

For more information about enabling the feature through CNC Console, see PCF Session Management.

- **Enable using REST API:** Set the Keys and Values for **Advanced Settings** under PCF Session Management Service. For more information about enabling the feature through REST API, see "PCF Session Management" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

You can configure the concurrency handling through Bulwark functionality using the CNC Console or REST API for Policy.

- **Configure using CNC Console:** Perform the feature configurations under the **Advanced Settings** section on the **PCF Session Management** page. For more information about configuring audit service, see PCF Session Management.
- **Configure using REST API:**Set the Keys and Values for **Advanced Settings** under PCF Session Management Service. For more information about enabling the feature through REST API, see "PCF Session Management" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

4.89.2 Support for Concurrency Handling using Bulwark Service in AM

The Bulwark Service integrates with the AM service to handle the concurrent requests for the AM Create, AM Update, AM Delete, and AM Update-Notify procedures. The AM service integration with Bulwark service handles the concurrent requests for the same subscriber in a concurrent and efficient manner.

In case of multiple create, update, or delete requests for the same subscriber, the AM service requests a lock for the SUPI or GPSI value from the Bulwark service. Once the lock is acquired, the AM create, AM Update, AM delete, or AM Update-Notify request is executed for the SUPI or GPSI value with which the lock is requested. After the successful completion of the requests, the AM service sends a request to release the lock to Bulwark.

As the notification is locked, pending transaction comes into picture.

When Bulwark-Service is Enabled, the following scenarios can occur between pending transaction and AM Update:

Table 4-47 Pending Transaction and Concurrency in AM Update

Pending Transaction	Concurrency in AM Update	Use-Case
Enabled	Disabled	Single Key - Session level lock will be acquired. Pending-transaction call flow would work.
Disabled	Enabled	Single Key - Subscriber level lock will be acquired. Update-Notify call flow with concurrency would work.

Table 4-47 (Cont.) Pending Transaction and Concurrency in AM Update

Pending Transaction	Concurrency in AM Update	Use-Case
Enabled	Enabled	Multi-Key - Session Level & Subscriber Level locks will be acquired.

When Bulwark-Service is Enabled, the following scenarios can occur between pending transaction and update-notify:

Table 4-48 Pending Transaction and Concurrency in AM Update-Notify

Pending Transaction	Concurrency in Update-Notify	Use-Case
Enabled	Disabled	Single Key - Session level lock will be acquired. Pending-transaction call flow would work.
Disabled	Enabled	Single Key - Subscriber level lock will be acquired. AM-Update call flow would work.
Enabled	Enabled	Multi-Key - Session Level & Subscriber Level locks will be acquired.

When Bulwark-Service is Enabled, the following scenarios can occur between pending transaction and AM Update and AM update-notify:

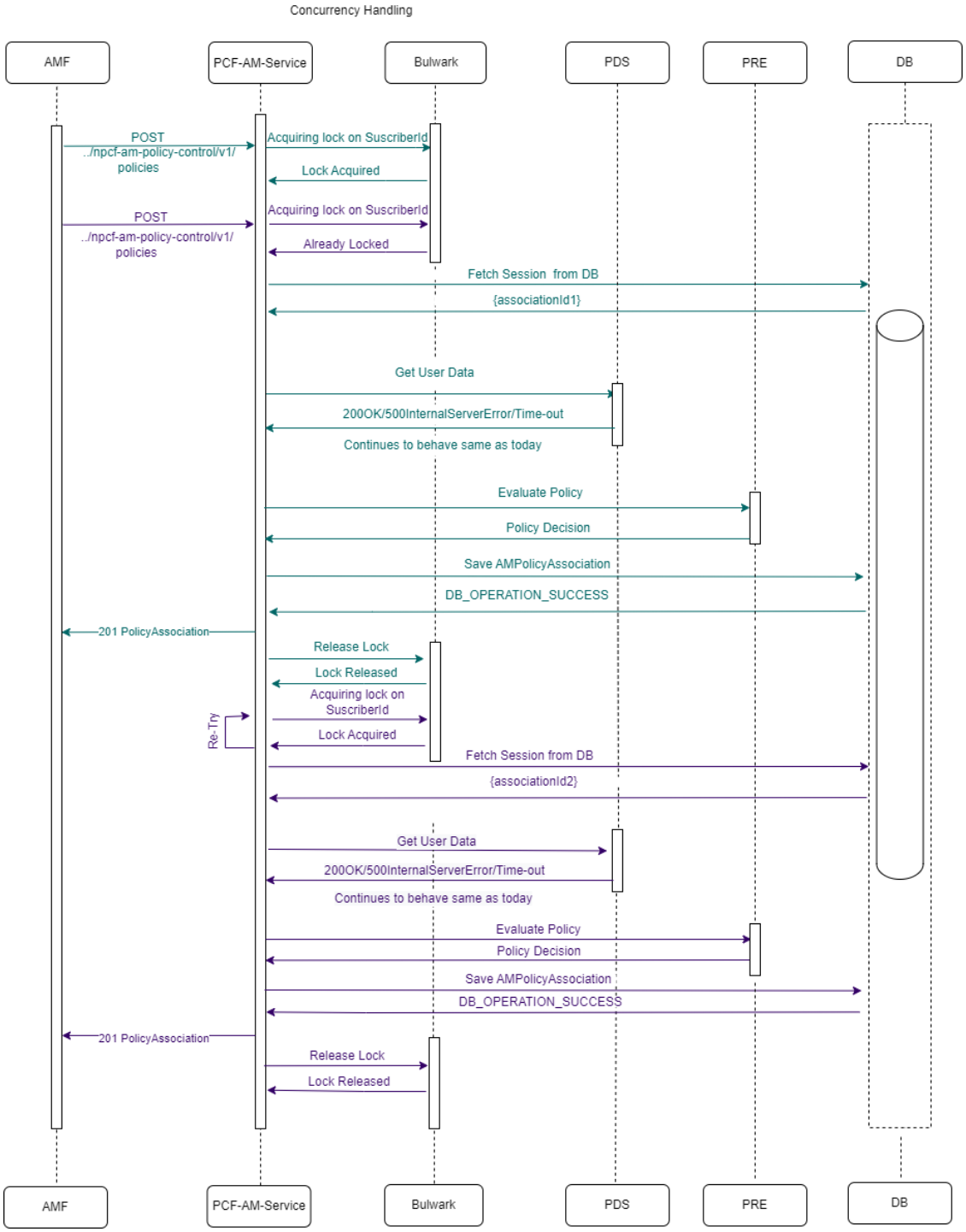
Table 4-49 Pending Transaction and Concurrency in AM Update as well as AM Update-Notify

Pending Transaction	Concurrency in AM Update	Concurrency in AM Update-Notify	Use-Case
Enabled	Disabled	Disabled	Session Level Lock Acquired - during collision, lock acquisition fails at SessionID
Disabled	Enabled	Disabled	Subscriber Level lock acquired - during collision, lock acquisition fails at subscriberID
Disabled	Enabled	Enabled	Subscriber Level lock acquired - during collision, lock acquisition fails at subscriberID
Enabled	Enabled	Disabled	Session Level & Subscriber Level locks will be acquired - during collision, lock acquisition will fail at SessionID

Call Flow for handling concurrent requests for AM Service

Call flow for two AM-Create requests for same subscriber and retry success

Figure 4-122 Call flow for two AM-Create requests for same subscriber and retry success



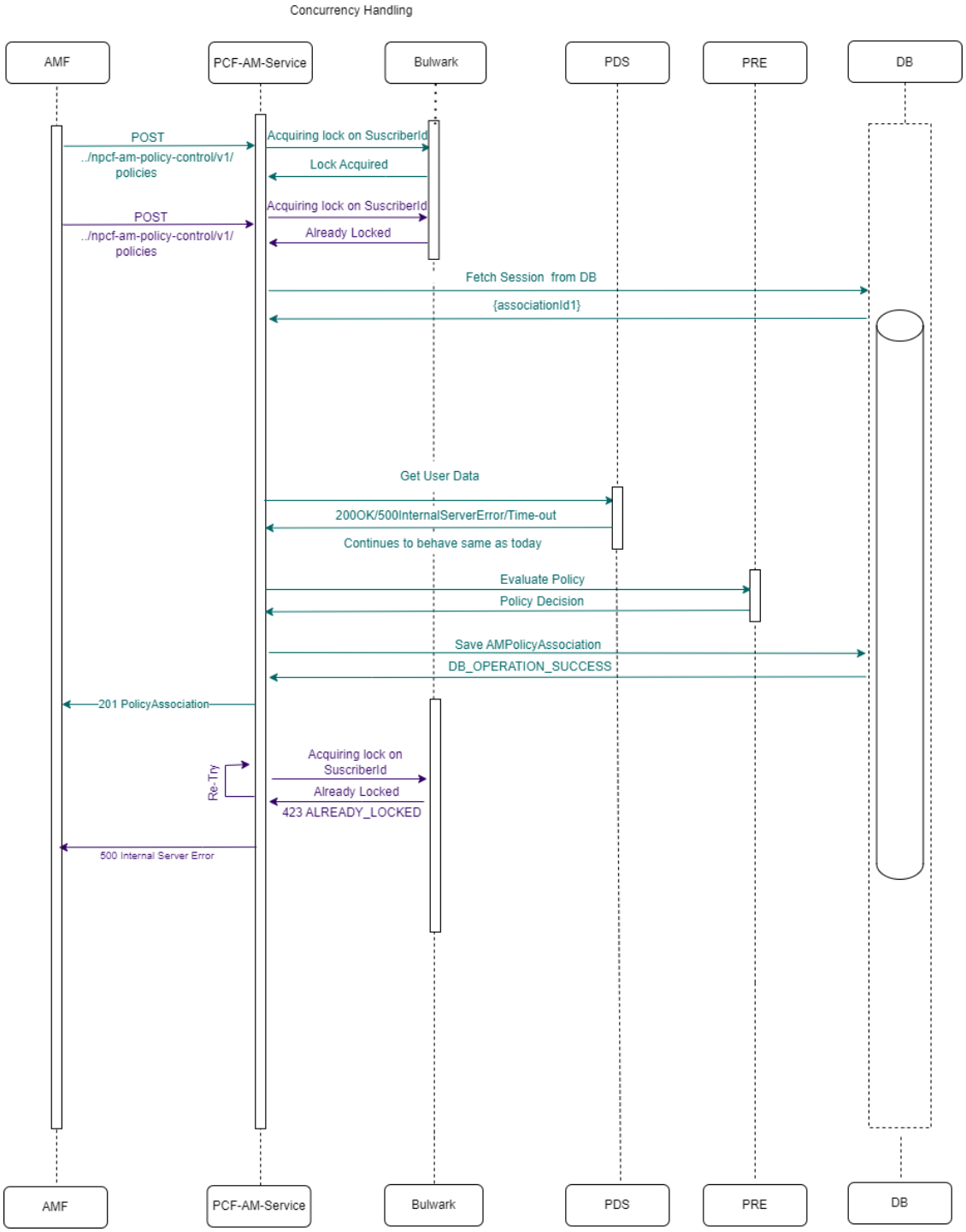
In this case, only concurrency is enabled for AM Create. AM Service acquires a single key lock using what is configured in RESOURCEID.SUFFIXLIST as a default is SUPI.

1. AM Service receives an AM Create request from AMF.

2. AM Service sends a lock request to Bulwark Service based on the configuration of RESOURCEID.SUFFIXLIST.
3. AM Service acquires a lock from Bulwark Service.
4. AM Service fetches the session details from the database and sends the create request to PDS.
5. AM service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
6. AM policy association is saved in the database.
7. After the successful completion of the create, AM Service sends a unlock release request to Bulwark Service to release the lock.

Two AM-Create requests for same Subscriber and retry failure (retry =1, CONCURRENCY.N15.CREATE.ALLOW_ON_SERVICE_FAILURE = false)

Figure 4-123 Two AM-Create requests for same Subscriber and retry failure (retry =1, CONCURRENCY.N15.CREATE.ALLOW_ON_SERVICE_FAILURE = false)



Concurrency is enabled for AM Create. AM Service acquires a single key lock using what is configured in RESOURCEID.SUFFIXLIST as a default is SUPI.

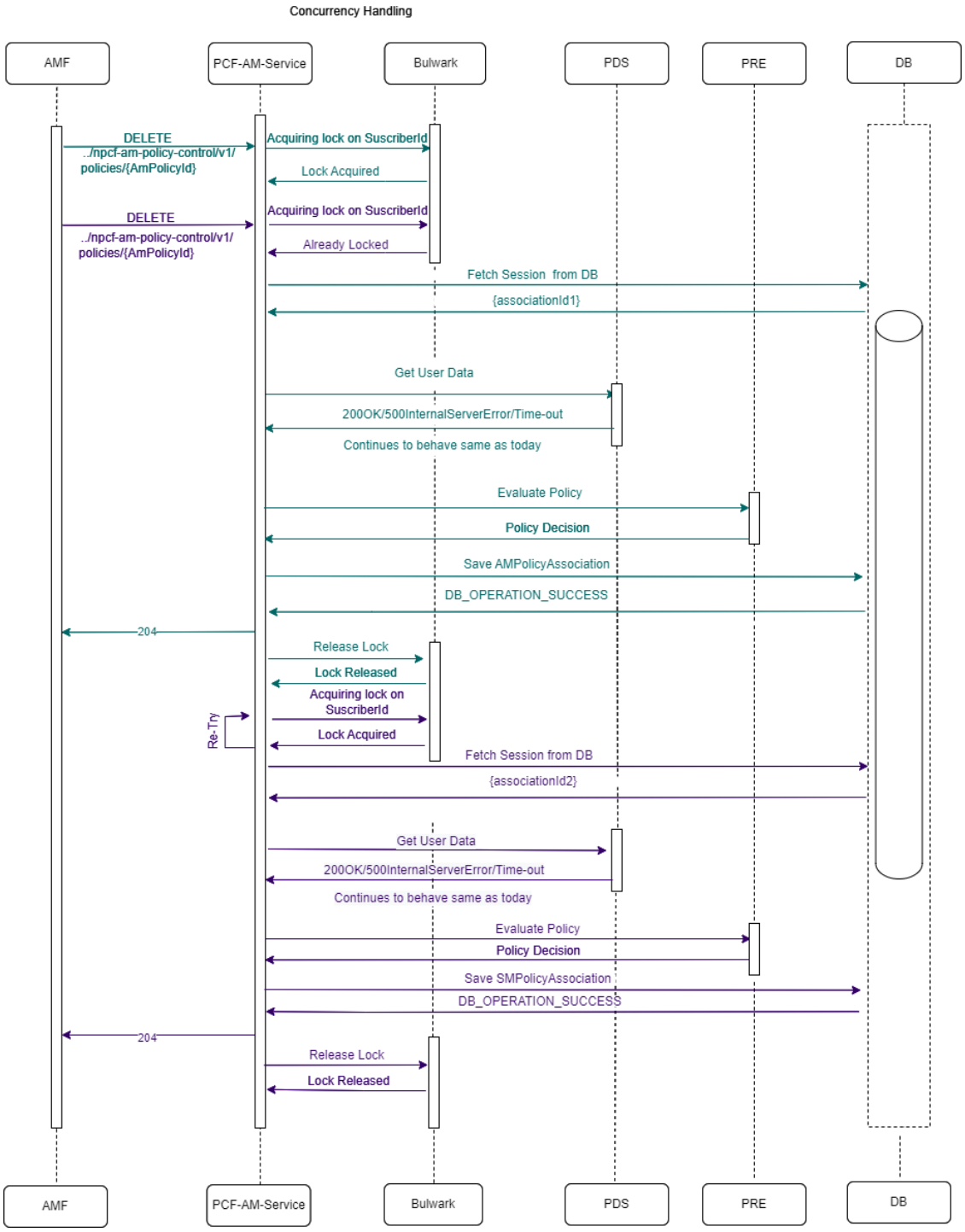
1. AM Service receives the first AM Create request from AMF.

2. AM Service sends a lock request to Bulwark Service based on what is configured in RESOURCEID.SUFFIXLIST.
3. AM Service acquires a lock for first request from Bulwark Service.
4. AM Service receives second AM Create request from AMF for the same subscriber.
5. AM Service sends a lock request to Bulwark Service for the second AM Create request.
6. Bulwark Service responds to AM Service for the second request with a 423 already locked message.
7. AM Service fetches the session details from the database for the first create request and sends the create request to PDS.
8. AM service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
9. AM policy association is saved in the database.
10. After the successful completion of the request, AM Service sends a lock release request to Bulwark Service to release the lock for the first create request.
11. AM Service retries to acquire the lock for the second create request from Bulwark Service.
12. AM Service acquires a lock for the second create request, processes the create request.
13. After completing the create for the second request, AM Service sends an unlock request to Bulwark Service to release the lock for the second request.

> **Note**
>
> If CONCURRENCY.N15.CREATE.ALLOW_ON_SERVICE_FAILURE is true and the request is allowed to be processed further even after exhausting the retries, the response could be 201 Success.

Two AM-Delete requests for same Subscriber and retry success

Figure 4-124 Two AM-Delete requests for same Subscriber and retry success



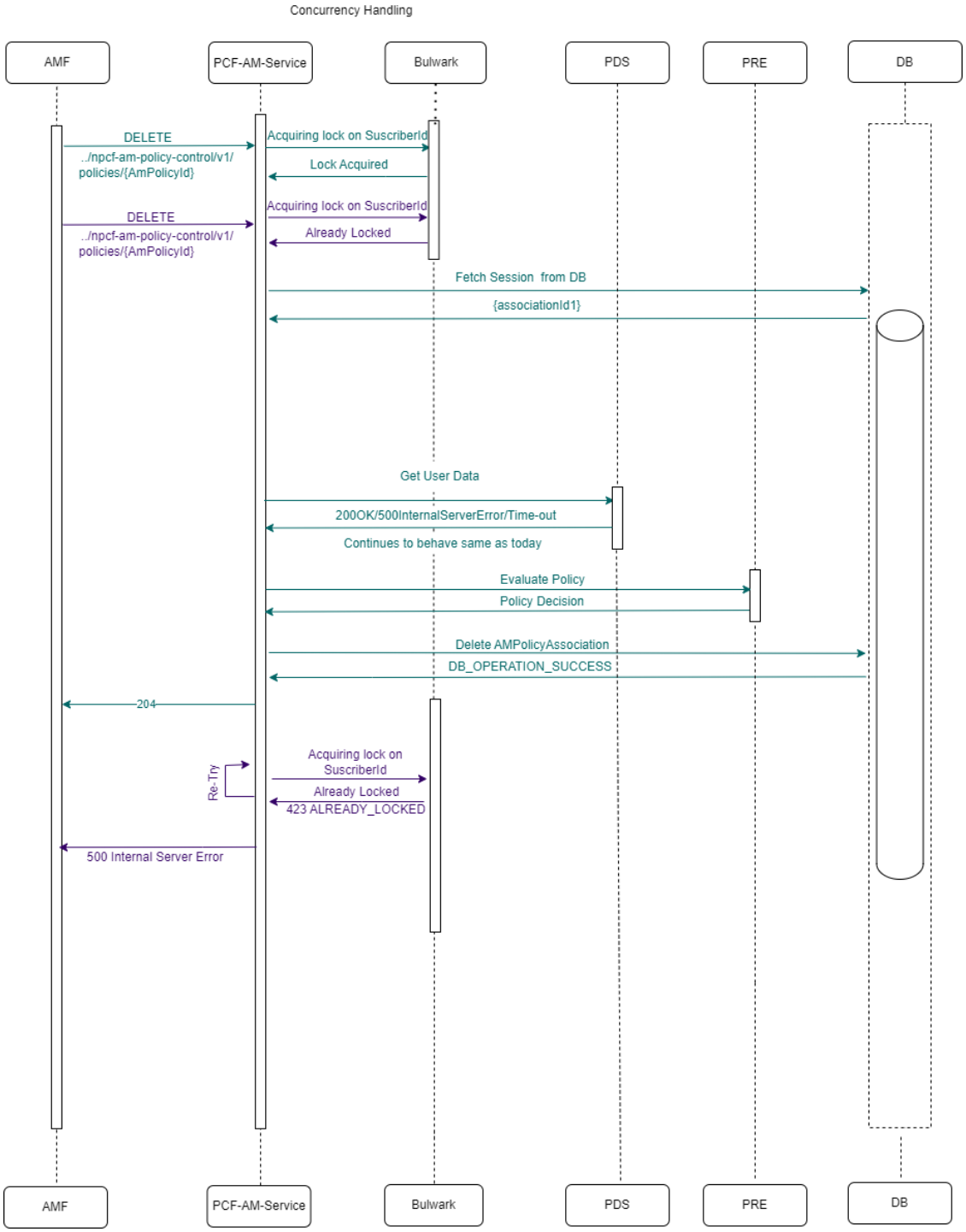
In this case, only concurrency is enabled for AM Delete. AM Service acquires a single key lock using what is configured in RESOURCEID.SUFFIXLIST as a default is SUPI.

1. AM Service receives an AM Delete request from AMF.
2. AM Service sends a lock request to Bulwark Service based on the configuration of RESOURCEID.SUFFIXLIST.

3. AM Service acquires a lock from Bulwark Service.
4. AM Service fetches the session details from the database and sends the delete request to PDS.
5. AM service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
6. AM policy association is deleted in the database.
7. After the successful completion of the delete, AM Service sends a unlock release request to Bulwark Service to release the lock.

Two AM-Delete requests for same Subscriber and retry failure (retry =1, CONCURRENCY.N15.TERMINATE.ALLOW_ON_SERVICE_FAILURE = false)

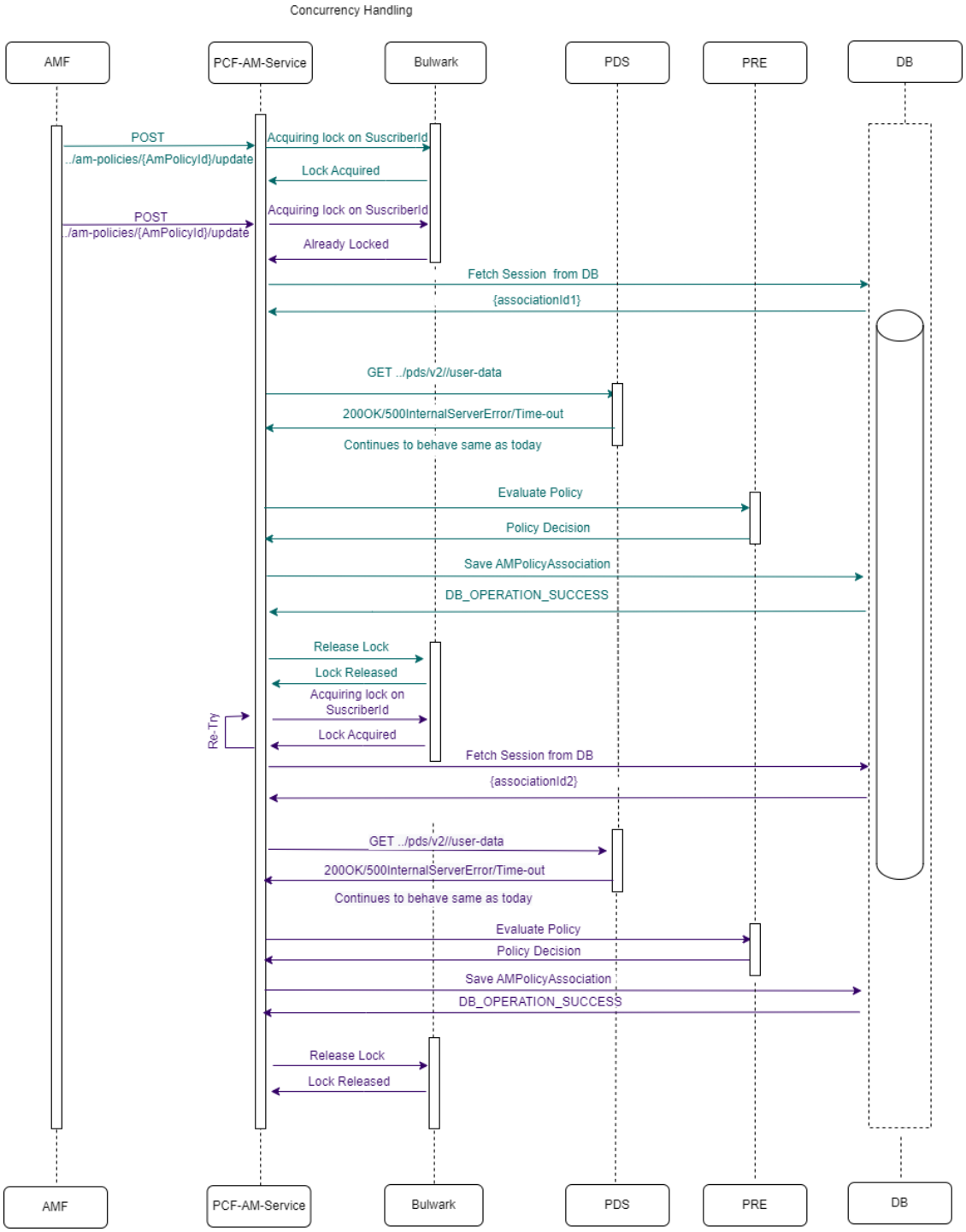
Figure 4-125 Two AM-Delete requests for same Subscriber and retry failure (retry =1, CONCURRENCY.N15.TERMINATE.ALLOW_ON_SERVICE_FAILURE = false)



> **Note**
>
> If CONCURRENCY.N15.TERMINATE.ALLOW_ON_SERVICE_FAILURE is true and the request is allowed to be processed further even after exhausting the retries, the response could be 201 Success.

Two AM-Update requests for same Subscriber but different Sessions and retry success

Figure 4-126 Two AM-Update requests for same Subscriber but different Sessions and retry success



In this case, only concurrency is enabled for AM Update. AM Service acquires a single key lock using what is configured in RESOURCEID.SUFFIXLIST as a default is SUPI.

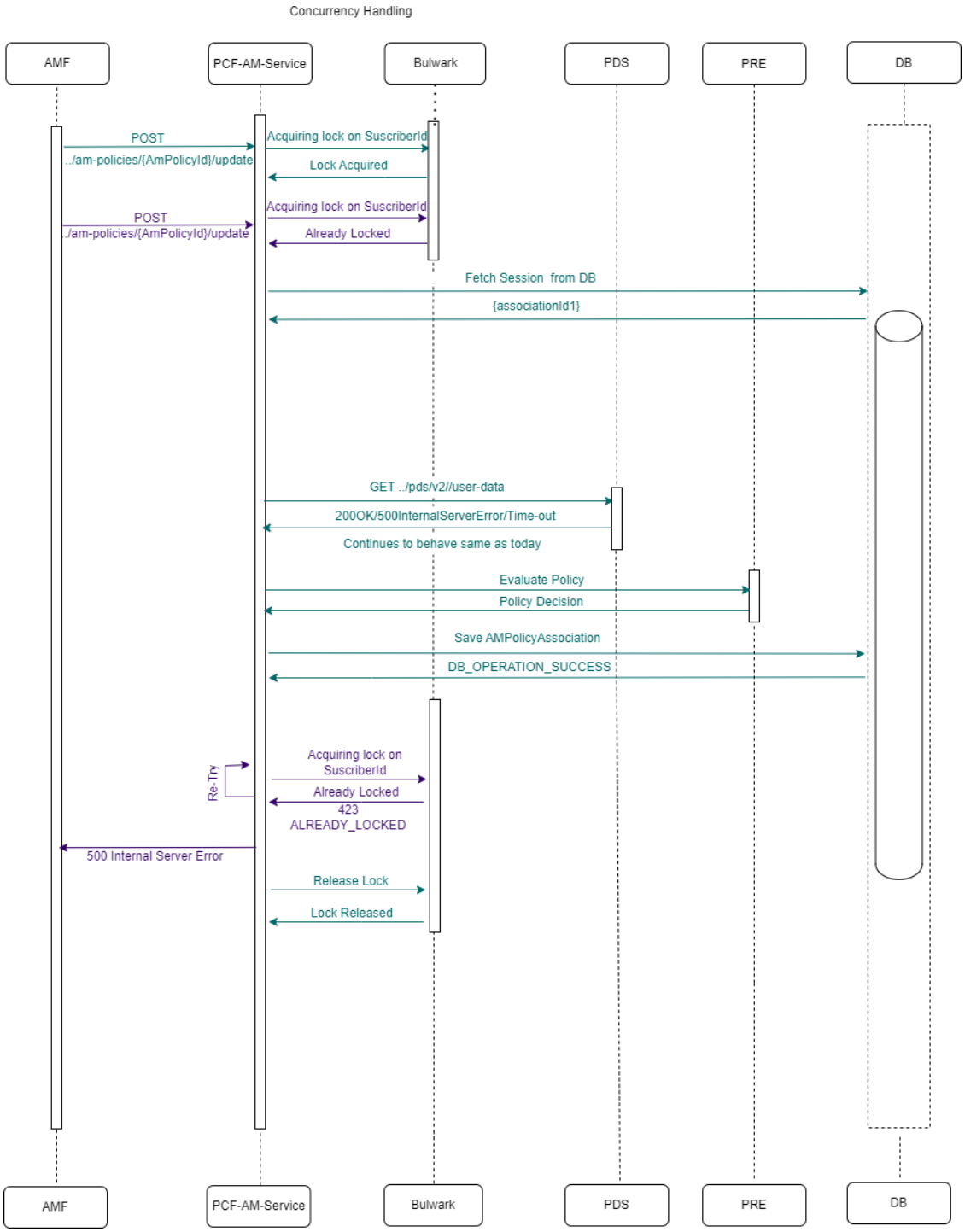
1. AM Service receives an AM Update request from AMF.

2. AM Service sends a lock request to Bulwark Service based on the configuration of RESOURCEID.SUFFIXLIST.
3. AM Service acquires a lock from Bulwark Service.
4. AM Service fetches the session details from the database and sends the update request to PDS.
5. AM service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
6. AM policy association is update in the database.
7. After the successful completion of the update, AM Service sends a unlock release request to Bulwark Service to release the lock.

Two AM-Update requests for same Subscriber but different Sessions and retry failure (retry =1, CONCURRENCY.N15.UPDATE.ALLOW_ON_SERVICE_FAILURE = false)

Figure 4-127 Two AM-Update requests for same Subscriber but different Sessions and retry failure (retry =1, CONCURRENCY.N15.UPDATE.ALLOW_ON_SERVICE_FAILURE =

false)



Concurrency is enabled for AM Update. AM Service acquires a single key lock using what is configured in RESOURCEID.SUFFIXLIST as a default is SUPI.

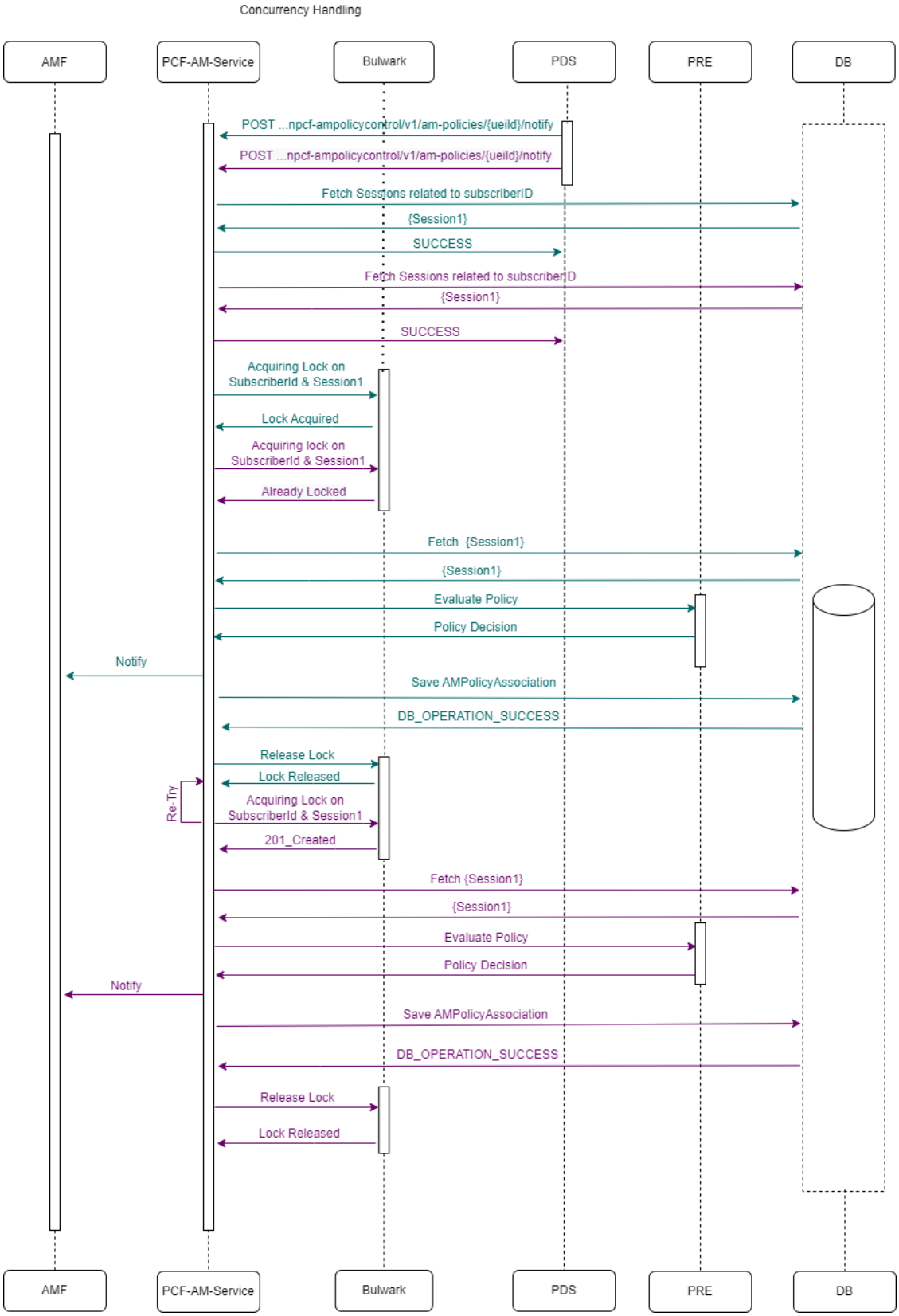
1. AM Service receives the first AM Update request from AMF.
2. AM Service sends a lock request to Bulwark Service based on what is configured in RESOURCEID.SUFFIXLIST.
3. AM Service acquires a lock for first request from Bulwark Service.

4. AM Service receives second AM Update request from AMF for the same subscriber.
5. AM Service sends a lock request to Bulwark Service for the second AM Update request.
6. Bulwark Service responds to AM Service for the second request with a 423 already locked message.
7. AM Service fetches the session details from the database for the first update request and sends the update request to PDS.
8. AM service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
9. AM policy association is saved in the database.
10. After the successful completion of the request, AM Service sends a lock release request to Bulwark Service to release the lock for the first update request.
11. AM Service retries to acquire the lock for the second update request from Bulwark Service.
12. AM Service acquires a lock for the second update request, processes the update request.
13. After completing the update for the second request, AM Service sends an unlock request to Bulwark Service to release the lock for the second request.

> **Note**
>
> If CONCURRENCY.N15.UPDATE.ALLOW_ON_SERVICE_FAILURE is true and the request is allowed to be processed further even after exhausting the retries, the response could be 201 Success.

Two AM-Notify requests for same Subscriber each having 1 Session

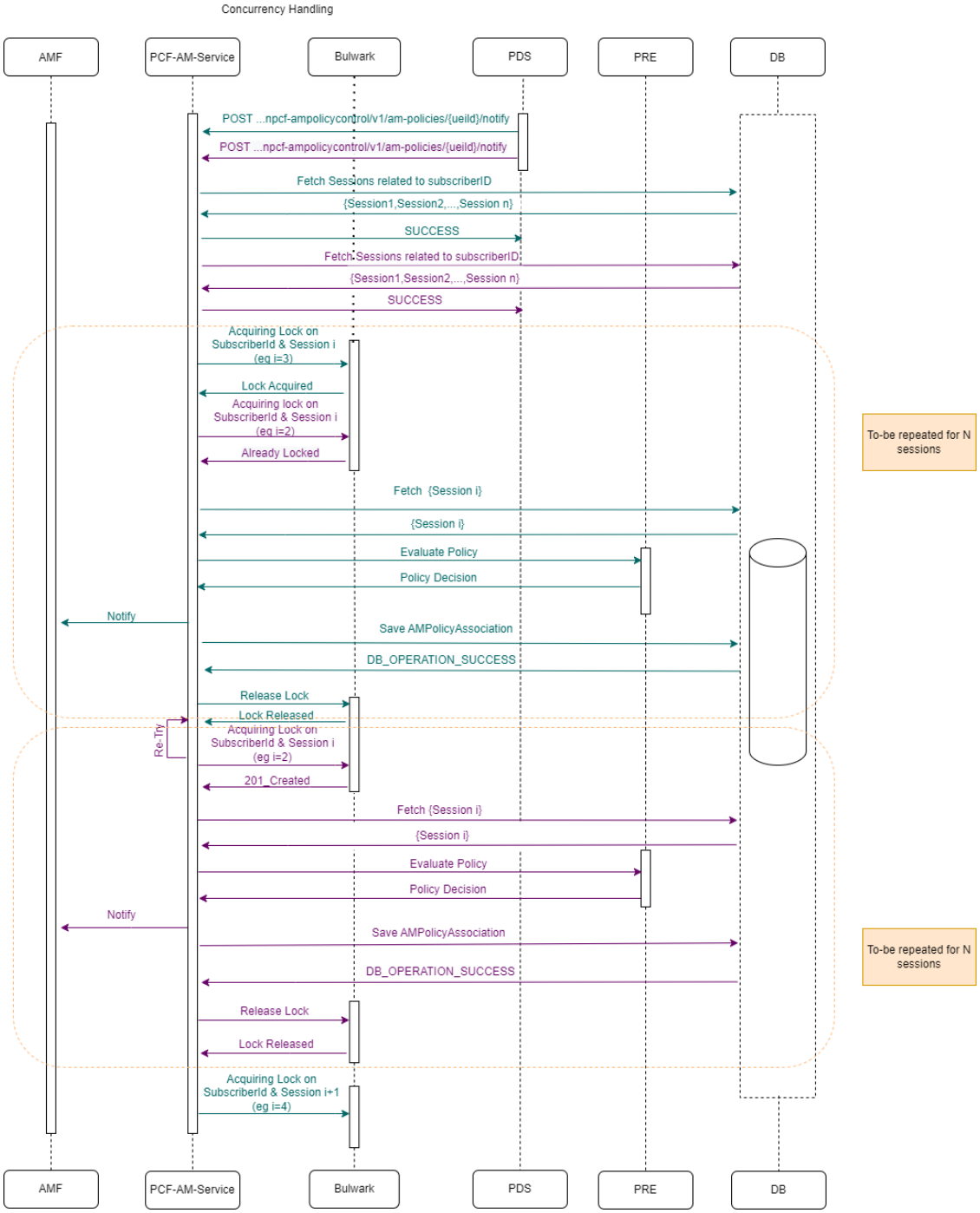
Figure 4-128 Two AM-Notify requests for same Subscriber each having 1 Session



1. Notify request will be received at AM service from PDS.
2. AM service will fetch the Sessions related to the subscriber ID
3. AM will send request to Bulwark for lock acquisition. For lock acquisition, We will get Lock on subscriber ID & Session ID iteratively for **N** number of sessions.
4. Once the lock is acquired, AM service will again send a request to fetch sessions, in case it got updated.
5. Once we have the lock & updated sessions, AM service will re-auth all the sessions sequentially.
6. Then AMPolicyAssociation will be saved in DB.
7. If the DB update is success, AM service will ask for Releasing lock of Bulwark Service.
8. After lock is released successfully, the update-notify request will be success.

Two AM-Notify request for same Subscriber each having n Sessions

Figure 4-129 Two AM-Notify request for same Subscriber each having n Sessions



Concurrency is enabled for AM Notify. AM Service acquires a single key lock using what is configured in RESOURCEID.SUFFIXLIST as a default is SUPI.

1. PDS sends a notify request to AM service.
2. AM service fetches the corresponding sessions from database and sends a success response to PDS.
3. A notify request tries to acquire the lock from bulwark service and gets the lock.

4. The notify request starts processing by fetching each sessions one by one.
5. AM service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
6. AM policy association is saved in DB.
7. On successful save, a notify response is sent to AMF and the lock is released.
8. This process gets repeated for n sessions of the Notify request.

Managing Concurrency Handling for AM Service

Enable

You can enable the concurrency handling through Bulwark using the CNC Console or REST API for Policy.

- **Enable using CNC Console:** Set value for the following parameters under the **Advanced Settings** section on the **PCF Access and Mobility** page:
  - CONCURRENCY.BULWARK_SERVICE_ENABLED
  - CONCURRENCY.N15.CREATE.ENABLED
  - CONCURRENCY.N15.CREATE.LOCK_LEASE_DURATION
  - CONCURRENCY.N15.CREATE.LOCK_WAIT_DURATION
  - CONCURRENCY.N15.CREATE.LOCK_REQUEST_RETRY_COUNT
  - CONCURRENCY.N15.CREATE.LOCK_REQUEST_RETRY_BACKOFF
  - CONCURRENCY.N15.TERMINATE.ENABLED
  - CONCURRENCY.N15.TERMINATE.LOCK_LEASE_DURATION
  - CONCURRENCY.N15.TERMINATE.LOCK_REQUEST_RETRY_BACKOFF
  - CONCURRENCY.N15.TERMINATE.LOCK_REQUEST_RETRY_COUNT
  - CONCURRENCY.N15.TERMINATE.LOCK_WAIT_DURATION
  - CONCURRENCY.N15.UPDATE.ENABLED
  - CONCURRENCY.N15.UPDATE.LOCK_LEASE_DURATION
  - CONCURRENCY.N15.UPDATE.LOCK_REQUEST_RETRY_BACKOFF
  - CONCURRENCY.N15.UPDATE.LOCK_REQUEST_RETRY_COUNT
  - CONCURRENCY.N15.UPDATE.LOCK_WAIT_DURATION
  - CONCURRENCY.N15.UPDATE_NOTIFY.ENABLED
  - CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_LEASE_DURATION
  - CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_REQUEST_RETRY_BACKOFF
  - CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_REQUEST_RETRY_COUNT
  - CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_WAIT_DURATION
  - RESOURCEID.SUFFIXLIST
  - ENABLE.ASSOCIATIONID.ENCODING
  - CONCURRENCY.N15.CREATE.ALLOW_ON_SERVICE_FAILURE
  - CONCURRENCY.N15.UPDATE.ALLOW_ON_SERVICE_FAILURE
  - CONCURRENCY.N15.UPDATE_NOTIFY.ALLOW_ON_SERVICE_FAILURE

- CONCURRENCY.N15.TERMINATE.ALLOW_ON_SERVICE_FAILURE

> **Note**
>
> If you do not make these parameters as configurable, the AM service uses the default values for these parameters.

For more information about enabling the feature through CNC Console, see PCF Access and Mobility .

- **Enable using REST API:** Set the Keys and Values for **Advanced Settings** under PCF Session Management Service. For more information about enabling the feature through REST API, see "Access and Mobility Service" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

You can configure the concurrency handling through Bulwark functionality using the CNC Console or REST API for Policy.

- **Configure using CNC Console:** Perform the feature configurations under the **Advanced Settings** section on the **PCF Access and Mobility** page. For more information about configuring audit service, see PCF Access and Mobility.
- **Configure using REST API:**Set the Keys and Values for **Advanced Settings** under PCF Session Management Service. For more information about enabling the feature through REST API, see "Access and Mobility Service" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

4.89.3 Support for Concurrency Handling using Bulwark Service in UE

The Bulwark Service integrates with the UE service to handle the concurrent requests for the UE Create, UE Update, UE Delete, and UE Update-Notify procedures. The UE service integration with Bulwark service handles the concurrent requests for the same subscriber in a concurrent and efficient manner.

In case of multiple create, update, or delete requests for the same subscriber, the UE service requests a lock for the SUPI or GPSI value from the Bulwark service. Once the lock is acquired, the UE create, UE Update, UE delete, or UE Update-Notify request is executed for the SUPI or GPSI value with which the lock is requested. After the successful completion of the requests, the UE service sends a request to release the lock to Bulwark.

As the notification is locked, pending transaction comes into picture.

When Bulwark-Service is Enabled, the following scenarios can occur between pending transaction and UE Update:

Table 4-50 Pending Transaction and Concurrency in UE Update

Pending Transaction	Concurrency in UE Update	Use-Case
Enabled	Disabled	Single Key - Session level lock will be acquired. Pending-transaction call flow would work.

Table 4-50 (Cont.) Pending Transaction and Concurrency in UE Update

Pending Transaction	Concurrency in UE Update	Use-Case
Disabled	Enabled	Single Key - Subscriber level lock will be acquired. Update-Notify call flow with concurrency would work.
Enabled	Enabled	Multi-Key - Session Level & Subscriber Level locks will be acquired.

When Bulwark-Service is Enabled, the following scenarios can occur between pending transaction and update-notify:

Table 4-51 Pending Transaction and Concurrency in UE Update-Notify

Pending Transaction	Concurrency in Update-Notify	Use-Case
Enabled	Disabled	Single Key - Session level lock will be acquired. Pending-transaction call flow would work.
Disabled	Enabled	Single Key - Subscriber level lock will be acquired. UE-Update call flow would work.
Enabled	Enabled	Multi-Key - Session Level & Subscriber Level locks will be acquired.

When Bulwark-Service is Enabled, the following scenarios can occur between pending transaction and UE Update and UE update-notify:

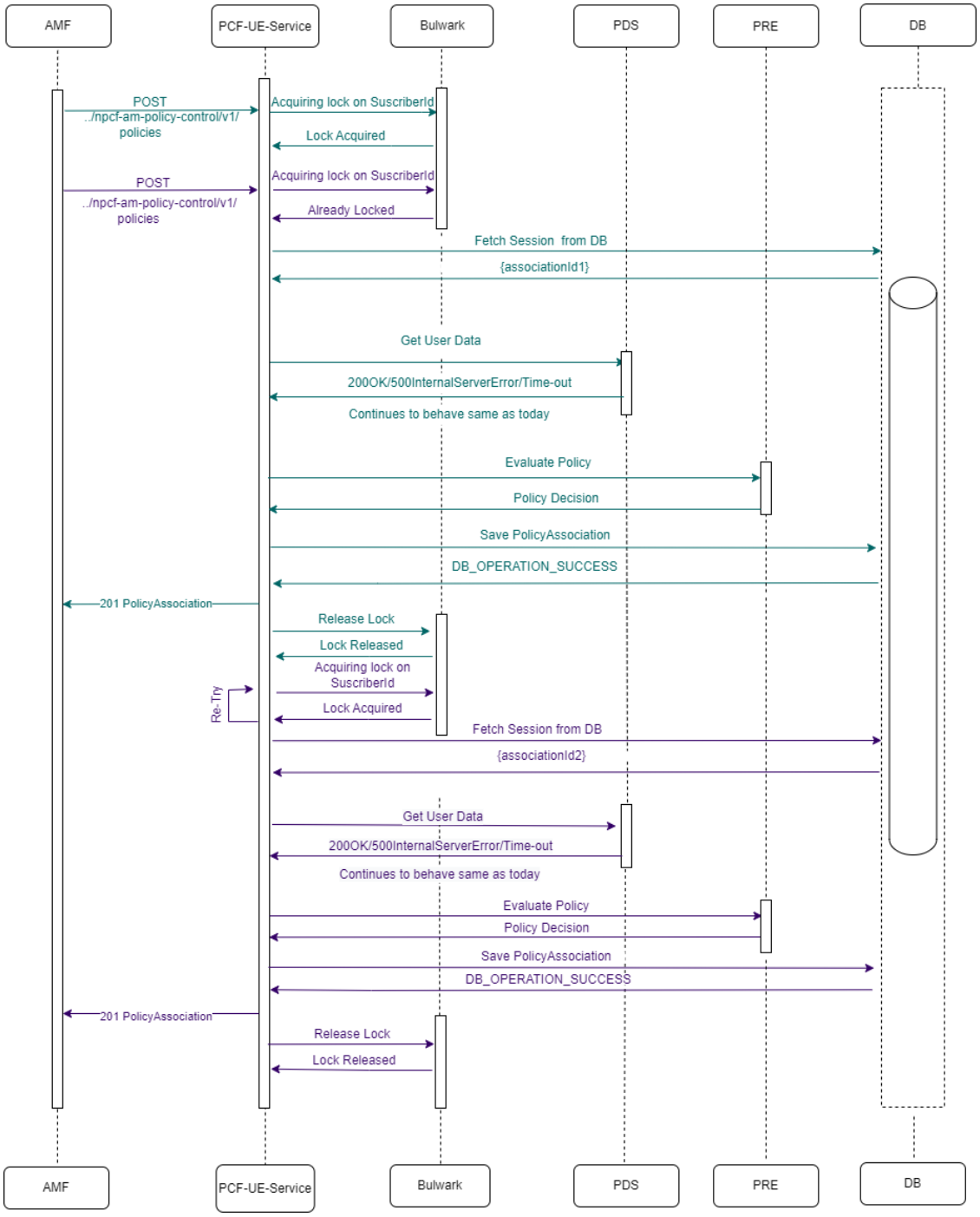
Table 4-52 Pending Transaction and Concurrency in UE Update as well as UE Update-Notify

Pending Transaction	Concurrency in UE Update	Concurrency in UE Update-Notify	Use-Case
Enabled	Disabled	Disabled	Session Level Lock Acquired - during collision, lock acquisition fails at SessionID
Disabled	Enabled	Disabled	Subscriber Level lock acquired - during collision, lock acquisition fails at subscriberID
Disabled	Enabled	Enabled	Subscriber Level lock acquired - during collision, lock acquisition fails at subscriberID
Enabled	Enabled	Disabled	Session Level & Subscriber Level locks will be acquired - during collision, lock acquisition will fail at SessionID

Call Flow for handling concurrent requests for UE Service

Two UE-Create requests for same Subscriber and retry success.

Figure 4-130 Two UE-Create requests for same Subscriber and retry success.

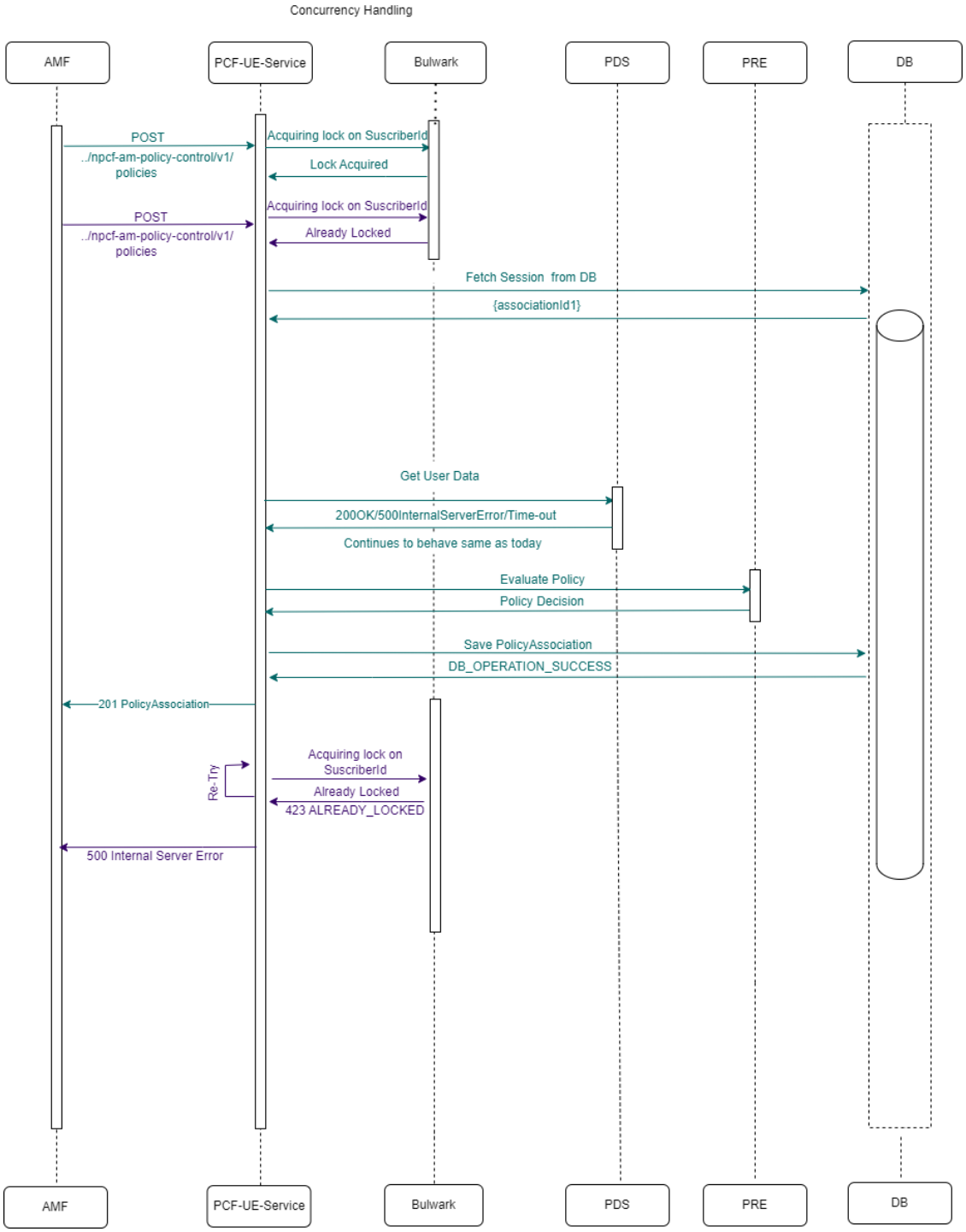


In this case, only concurrency is enabled for UE Create. UE Service acquires a single key lock using what is configured in RESOURCEID.SUFFIXLIST as a default is SUPI.

1. UE Service receives an UE Create request from AMF.
2. UE Service sends a lock request to Bulwark Service based on the configuration of RESOURCEID.SUFFIXLIST.
3. UE Service acquires a lock from Bulwark Service.
4. UE Service fetches the session details from the database and sends the create request to PDS.
5. UE service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
6. UE policy association is saved in the database.
7. After the successful completion of the create, UE Service sends a unlock release request to Bulwark Service to release the lock.

Two UE-Create requests for same Subscriber and retry failure (retry =1, CONCURRENCY.N15.CREATE.ALLOW_ON_SERVICE_FAILURE = false)

Figure 4-131 Two UE-Create requests for same Subscriber and retry failure (retry =1, CONCURRENCY.N15.CREATE.ALLOW_ON_SERVICE_FAILURE = false)



Concurrency is enabled for UE Create. UE Service acquires a single key lock using what is configured in RESOURCEID.SUFFIXLIST as a default is SUPI.

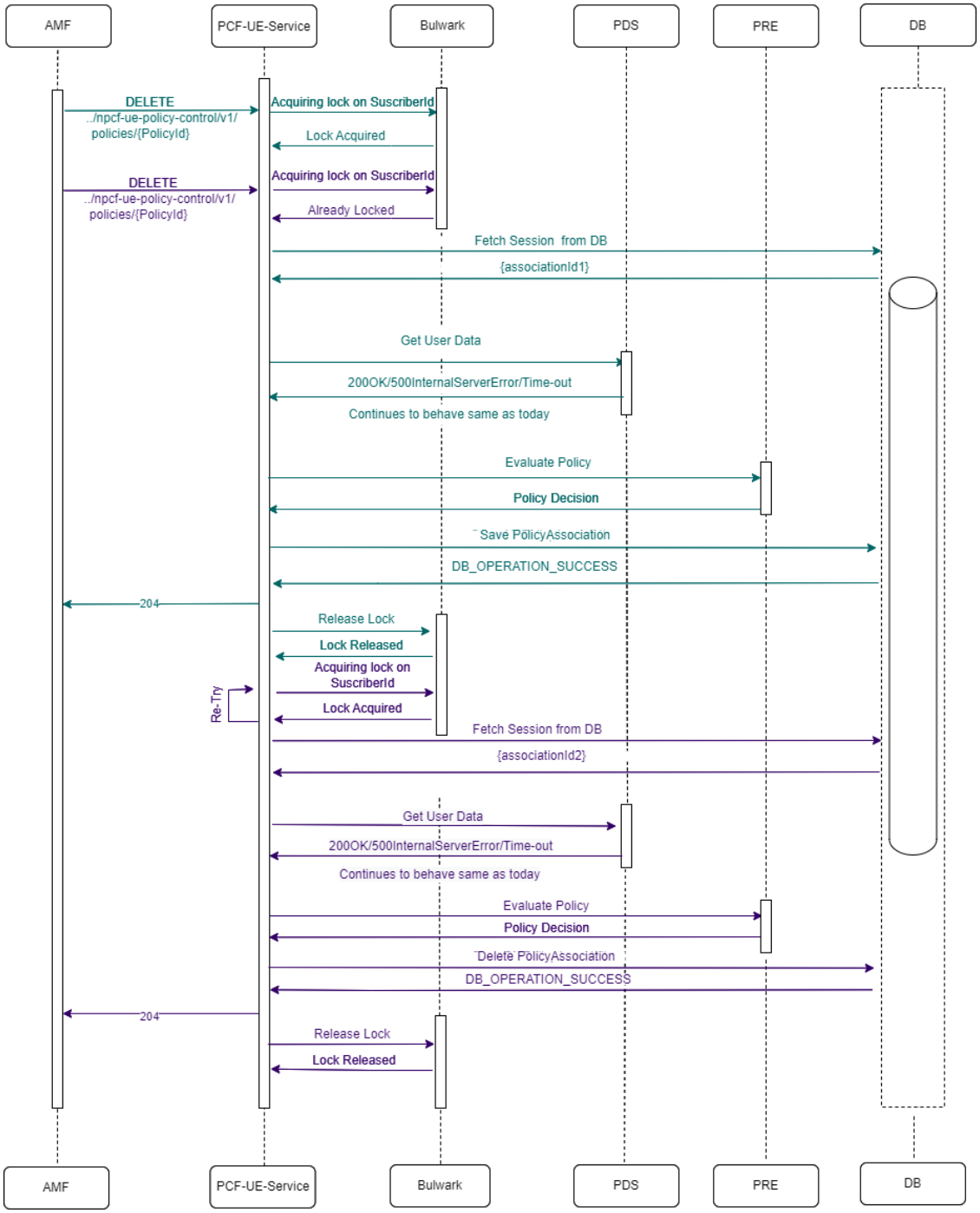
1. UE Service receives the first UE Create request from AMF.

2. UE Service sends a lock request to Bulwark Service based on what is configured in RESOURCEID.SUFFIXLIST.
3. UE Service acquires a lock for first request from Bulwark Service.
4. UE Service receives second UE Create request from AMF for the same subscriber.
5. UE Service sends a lock request to Bulwark Service for the second UE Create request.
6. Bulwark Service responds to UE Service for the second request with a 423 already locked message.
7. UE Service fetches the session details from the database for the first create request and sends the create request to PDS.
8. UE service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
9. UE policy association is saved in the database.
10. After the successful completion of the request, UE Service sends a lock release request to Bulwark Service to release the lock for the first create request.
11. UE Service retries to acquire the lock for the second create request from Bulwark Service.
12. UE Service acquires a lock for the second create request, processes the create request.
13. After completing the create for the second request, UE Service sends an unlock request to Bulwark Service to release the lock for the second request.

> **Note**
>
> If CONCURRENCY.N15.CREATE.ALLOW_ON_SERVICE_FAILURE is true and the request is allowed to be processed further even after exhausting the retries, the response could be 201 Success.

Two UE-Delete requests for same Subscriber and retry success.

Figure 4-132 Two UE-Delete requests for same Subscriber and retry success.



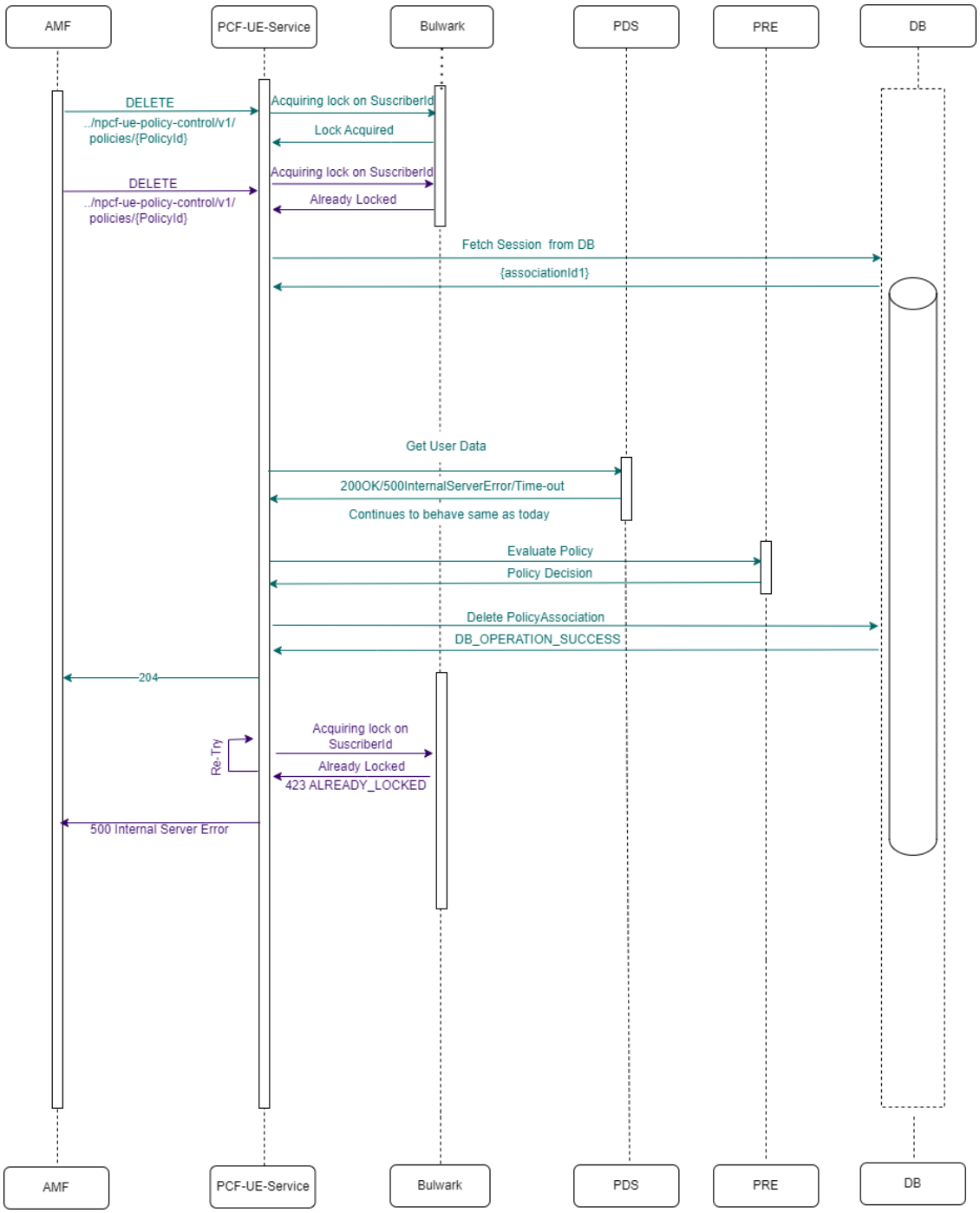
In this case, only concurrency is enabled for UE Delete. UE Service acquires a single key lock using what is configured in RESOURCEID.SUFFIXLIST as a default is SUPI.

1. UE Service receives an UE Delete request from AMF.
2. UE Service sends a lock request to Bulwark Service based on the configuration of RESOURCEID.SUFFIXLIST.
3. UE Service acquires a lock from Bulwark Service.

4. UE Service fetches the session details from the database and sends the delete request to PDS.
5. UE service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
6. UE policy association is deleted in the database.
7. After the successful completion of the delete, UE Service sends a unlock release request to Bulwark Service to release the lock.

Two UE-Delete requests for same Subscriber and retry failure (retry =1, CONCURRENCY.N15.TERMINATE.ALLOW_ON_SERVICE_FAILURE = false)

Figure 4-133 Two UE-Delete requests for same Subscriber and retry failure (retry =1, CONCURRENCY.N15.TERMINATE.ALLOW_ON_SERVICE_FAILURE = false)



> **Note**
>
> If CONCURRENCY.N15.CREATE.ALLOW_ON_SERVICE_FAILURE is true, as the request is allowed to be processed further even after exhausting the retries, the response could be 201 Success.

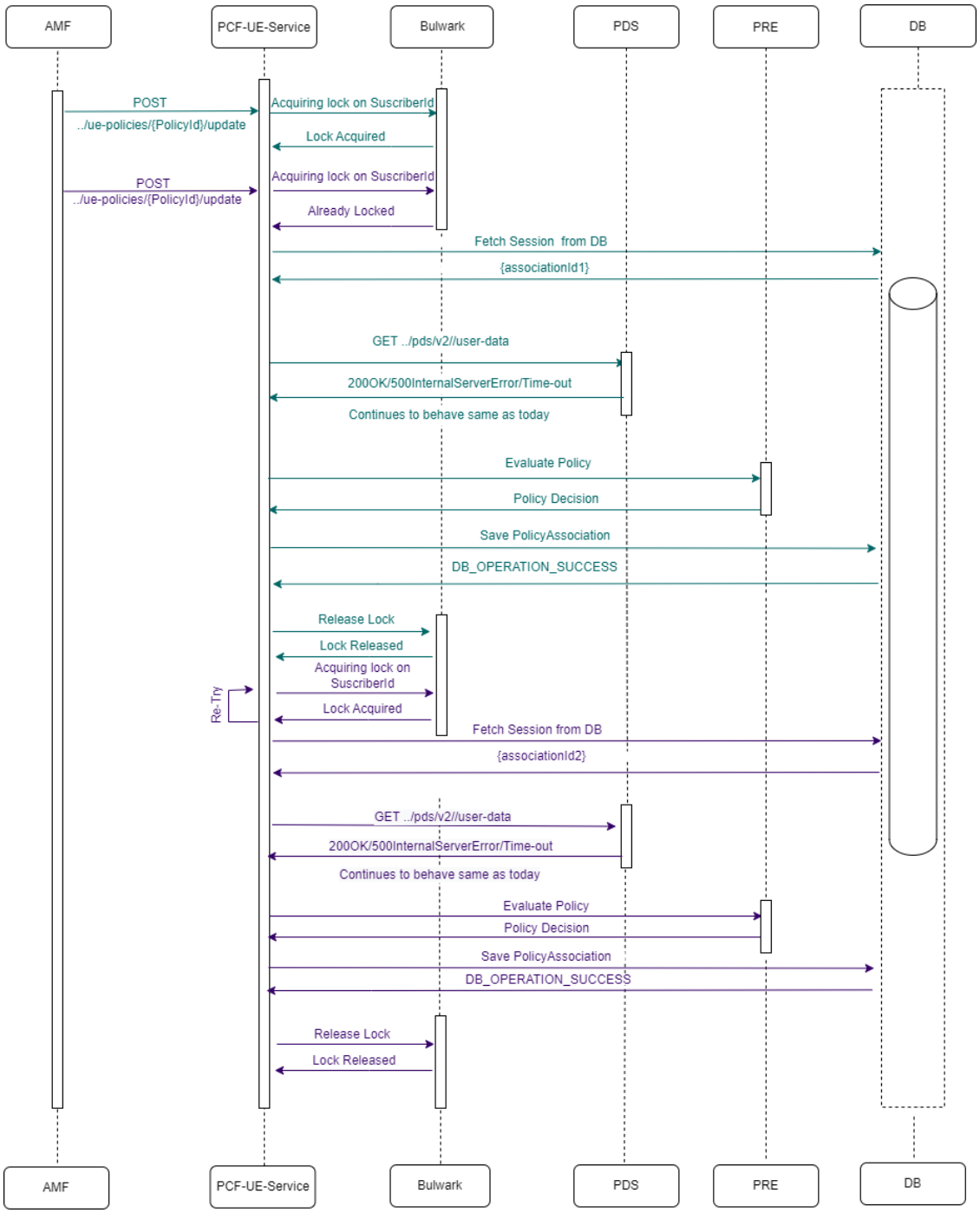
Concurrency is enabled for UE Delete. UE Service acquires a single key lock using what is configured in RESOURCEID.SUFFIXLIST as a default is SUPI.

1. UE Service receives the first UE Delete request from AMF.
2. UE Service sends a lock request to Bulwark Service based on what is configured in RESOURCEID.SUFFIXLIST.
3. UE Service acquires a lock for first request from Bulwark Service.
4. UE Service receives second UE Delete request from AMF for the same subscriber.
5. UE Service sends a lock request to Bulwark Service for the second UE Delete request.
6. Bulwark Service responds to UE Service for the second request with a 423 already locked message.
7. UE Service fetches the session details from the database for the first delete request and sends the delete request to PDS.
8. UE service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
9. UE policy association is saved in the database.
10. After the successful completion of the request, UE Service sends a lock release request to Bulwark Service to release the lock for the first delete request.
11. UE Service retries to acquire the lock for the second delete request from Bulwark Service.
12. UE Service acquires a lock for the second delete request, processes the delete request.
13. After completing the delete for the second request, UE Service sends an unlock request to Bulwark Service to release the lock for the second request.

> **Note**
>
> If CONCURRENCY.N15.TERMINATE.ALLOW_ON_SERVICE_FAILURE is true and the request is allowed to be processed further even after exhausting the retries, the response could be 201 Success.

Two UE-Update requests for same Subscriber but different Sessions and retry success.

Figure 4-134 Two UE-Update requests for same Subscriber but different Sessions and retry success.



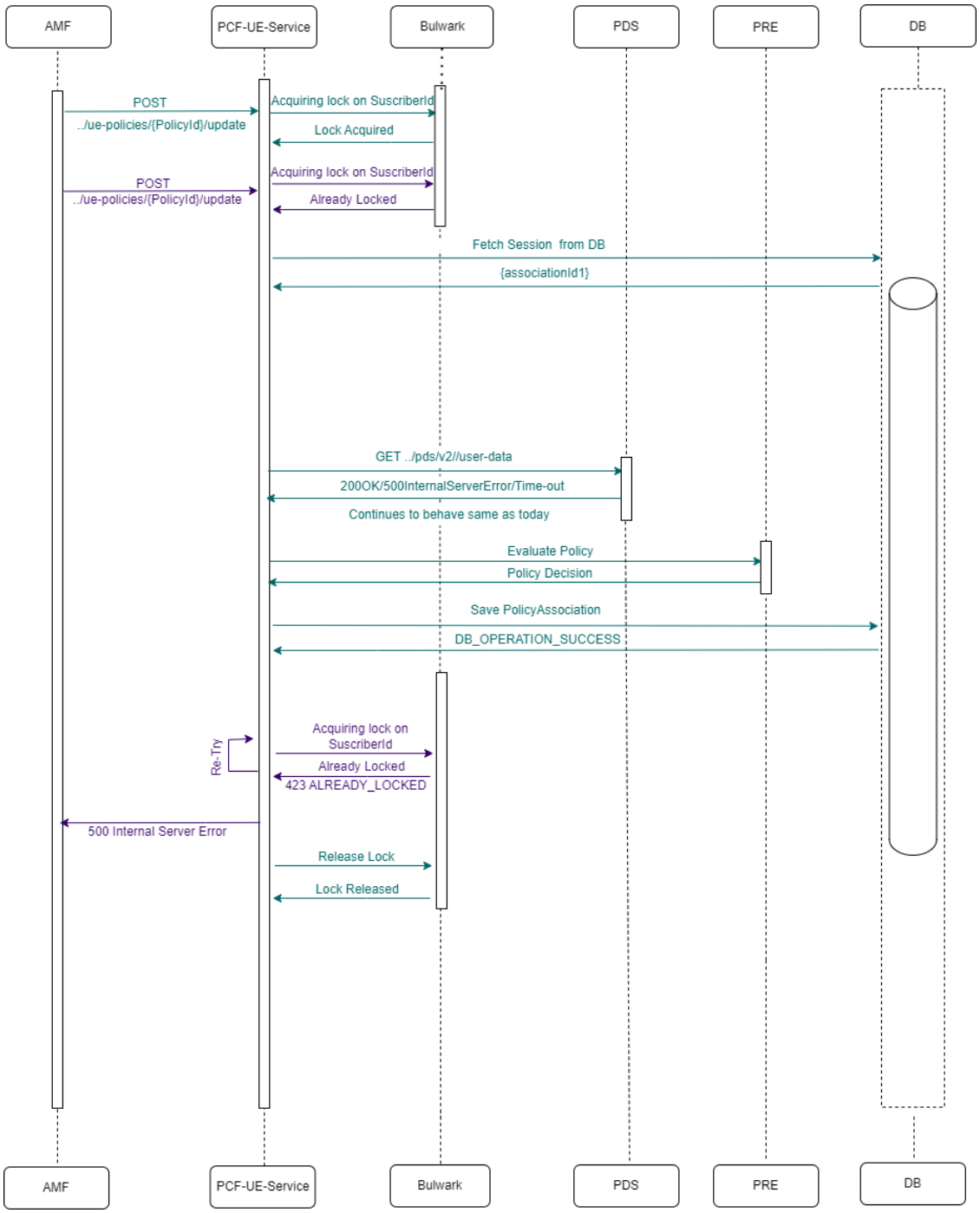
In this case, only concurrency is enabled for UE Update. UE Service acquires a single key lock using what is configured in RESOURCEID.SUFFIXLIST as a default is SUPI.

1. UE Service receives an UE Update request from AMF.
2. UE Service sends a lock request to Bulwark Service based on the configuration of RESOURCEID.SUFFIXLIST.
3. UE Service acquires a lock from Bulwark Service.

4. UE Service fetches the session details from the database and sends the update request to PDS.
5. UE service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
6. UE policy association is update in the database.
7. After the successful completion of the update, AM Service sends a unlock release request to Bulwark Service to release the lock.

Two UE-Update requests for same Subscriber but different Sessions and retry failure (retry =1, CONCURRENCY.N15.UPDATE.ALLOW_ON_SERVICE_FAILURE = false)

Figure 4-135 Two UE-Update requests for same Subscriber but different Sessions and retry failure (retry =1, CONCURRENCY.N15.UPDATE.ALLOW_ON_SERVICE_FAILURE = false)



Concurrency is enabled for UE Update. UE Service acquires a single key lock using what is configured in RESOURCEID.SUFFIXLIST as a default is SUPI.

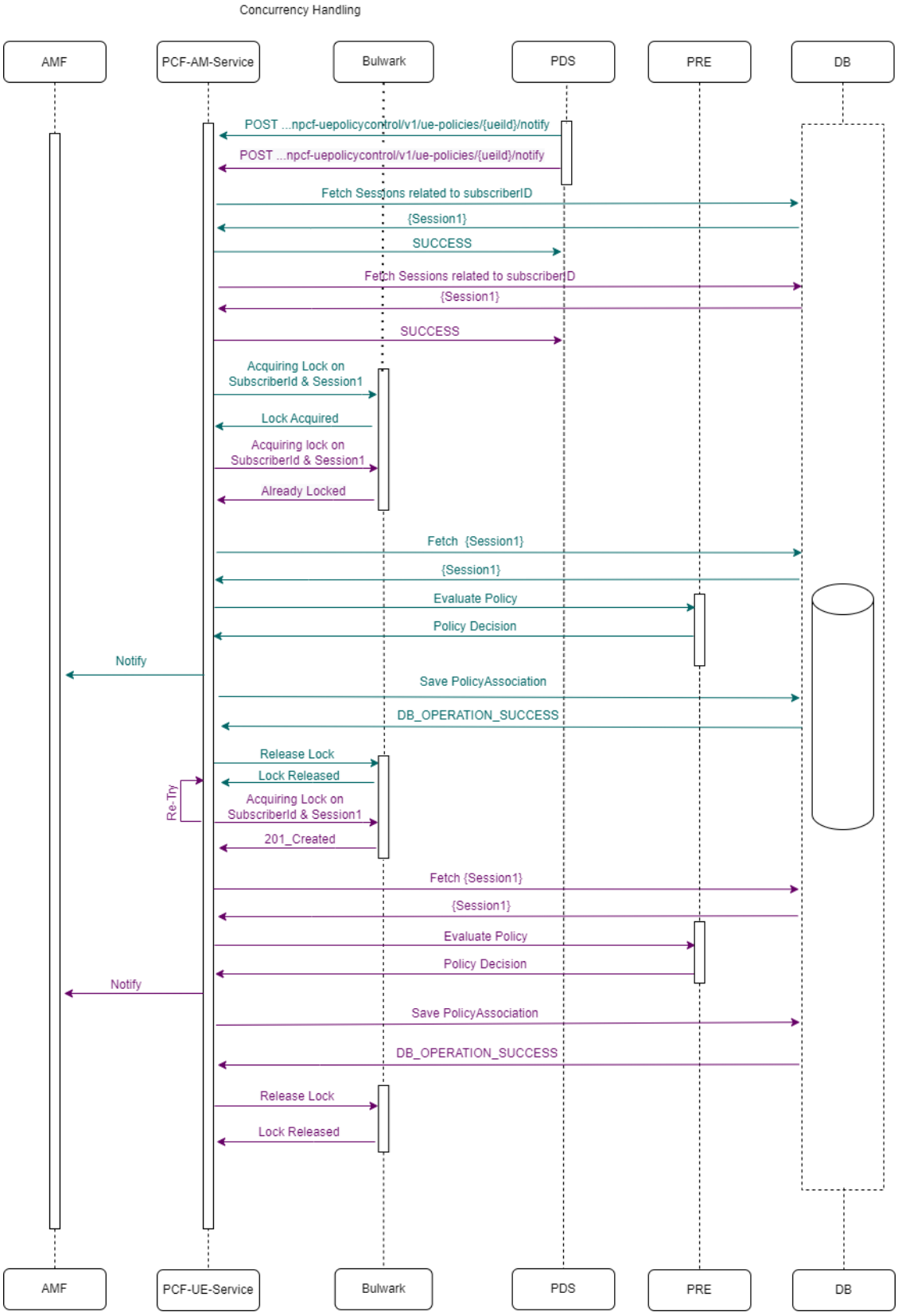
1. UE Service receives the first UE Update request from AMF.
2. UE Service sends a lock request to Bulwark Service based on what is configured in RESOURCEID.SUFFIXLIST.

3. UE Service acquires a lock for first request from Bulwark Service.
4. UE Service receives second UE Update request from AMF for the same subscriber.
5. UE Service sends a lock request to Bulwark Service for the second UE Update request.
6. Bulwark Service responds to UE Service for the second request with a 423 already locked message.
7. UE Service fetches the session details from the database for the first update request and sends the update request to PDS.
8. UE service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
9. UE policy association is saved in the database.
10. After the successful completion of the request, UE Service sends a lock release request to Bulwark Service to release the lock for the first update request.
11. UE Service retries to acquire the lock for the second update request from Bulwark Service.
12. UE Service acquires a lock for the second update request, processes the update request.
13. After completing the update for the second request, UE Service sends an unlock request to Bulwark Service to release the lock for the second request.

> **Note**
>
> If CONCURRENCY.N15.UPDATE.ALLOW_ON_SERVICE_FAILURE is true and the request is allowed to be processed further even after exhausting the retries, the response could be 201 Success.

Two UENotify requests for same Subscriber each having 1 session

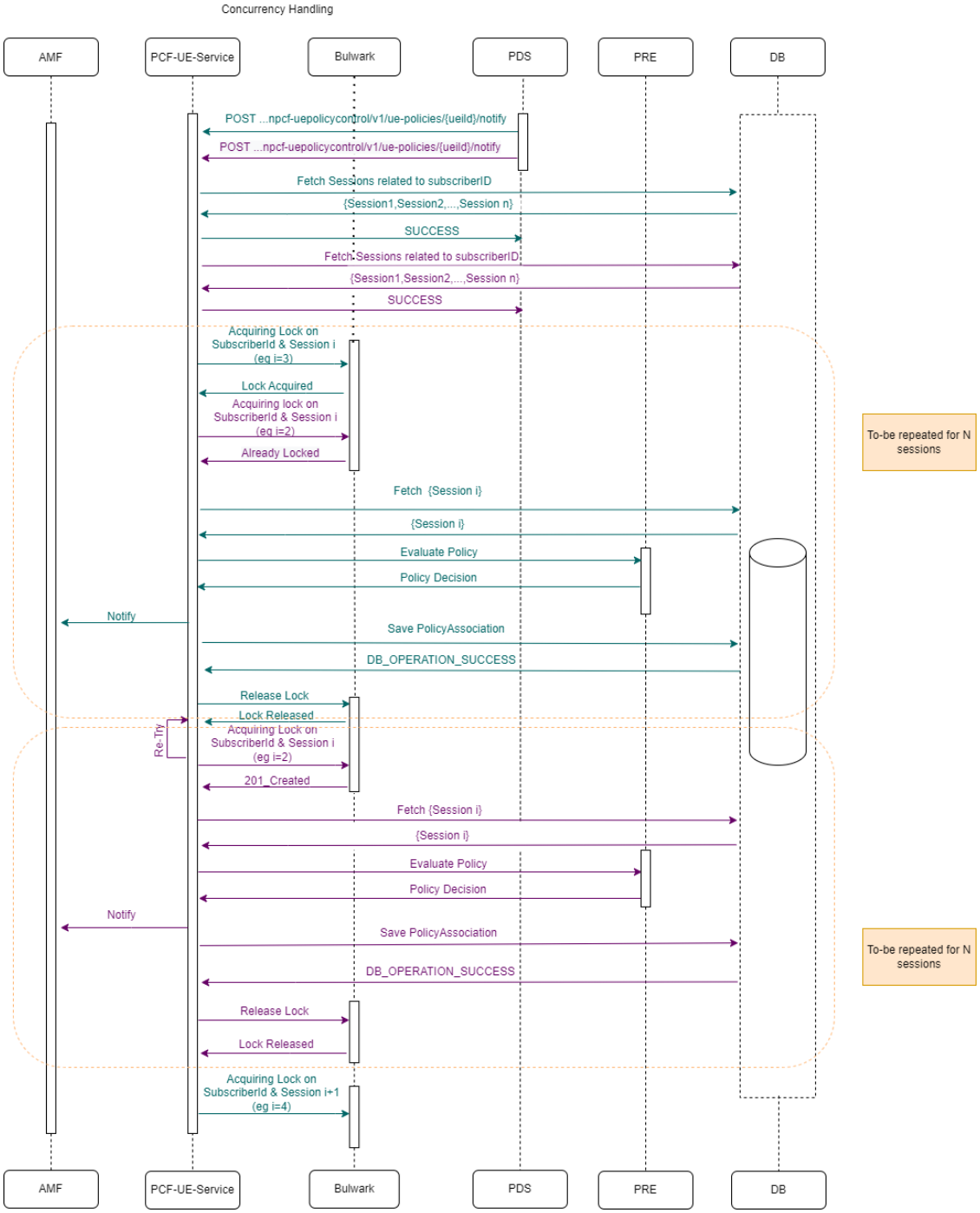
Figure 4-136 Two UENotify requests for same Subscriber each having 1 session



1. Notify request will be received at UE service from PDS.
2. UE service will fetch the Sessions related to the subscriber ID
3. UE will send request to Bulwark for lock acquisition. For lock acquisition, We will get Lock on subscriber ID & Session ID iteratively for **N** number of sessions.
4. Once the lock is acquired, UE service will again send a request to fetch sessions, in case it got updated.
5. Once we have the lock & updated sessions, UE service will re-auth all the sessions sequentially.
6. Then PolicyAssociation will be saved in DB.
7. If the DB update is success, UE service will ask for Releasing lock of Bulwark Service.
8. After lock is released successfully, the update-notify request will be success.

Two UENotify request for same Subscriber each having n sessions

Figure 4-137 Two UENotify request for same Subscriber each having n sessions



Concurrency is enabled for UE Notify. UE Service acquires a single key lock using what is configured in RESOURCEID.SUFFIXLIST as a default is SUPI.

1. PDS sends a notify request to UE service.
2. UE service fetches the corresponding sessions from database and sends a success response to PDS.
3. A notify request tries to acquire the lock from bulwark service and gets the lock.

4. The notify request starts processing by fetching each sessions one by one.
5. UE service sends a request to PRE for policy evaluation. PRE, in turn replies with the Policy Decision.
6. UE policy association is saved in DB.
7. On successful save, a notify response is sent to AMF and the lock is released.
8. This process gets repeated for n sessions of the Notify request.

Managing Concurrency Handling for UE Service

Enable

You can enable the concurrency handling through Bulwark using the CNC Console or REST API for Policy.

- **Enable using CNC Console:** Set value for the following parameters under the **Advanced Settings** section on the **PCF UE Policy Service** page:
  - CONCURRENCY.BULWARK_SERVICE_ENABLED
  - CONCURRENCY.N15.CREATE.ENABLED
  - CONCURRENCY.N15.CREATE.LOCK_LEASE_DURATION
  - CONCURRENCY.N15.CREATE.LOCK_WAIT_DURATION
  - CONCURRENCY.N15.CREATE.LOCK_REQUEST_RETRY_COUNT
  - CONCURRENCY.N15.CREATE.LOCK_REQUEST_RETRY_BACKOFF
  - CONCURRENCY.N15.TERMINATE.ENABLED
  - CONCURRENCY.N15.TERMINATE.LOCK_LEASE_DURATION
  - CONCURRENCY.N15.TERMINATE.LOCK_REQUEST_RETRY_BACKOFF
  - CONCURRENCY.N15.TERMINATE.LOCK_REQUEST_RETRY_COUNT
  - CONCURRENCY.N15.TERMINATE.LOCK_WAIT_DURATION
  - CONCURRENCY.N15.UPDATE.ENABLED
  - CONCURRENCY.N15.UPDATE.LOCK_LEASE_DURATION
  - CONCURRENCY.N15.UPDATE.LOCK_REQUEST_RETRY_BACKOFF
  - CONCURRENCY.N15.UPDATE.LOCK_REQUEST_RETRY_COUNT
  - CONCURRENCY.N15.UPDATE.LOCK_WAIT_DURATION
  - CONCURRENCY.N15.UPDATE_NOTIFY.ENABLED
  - CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_LEASE_DURATION
  - CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_REQUEST_RETRY_BACKOFF
  - CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_REQUEST_RETRY_COUNT
  - CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_WAIT_DURATION
  - RESOURCEID.SUFFIXLIST
  - ENABLE.ASSOCIATIONID.ENCODING
  - CONCURRENCY.N15.CREATE.ALLOW_ON_SERVICE_FAILURE
  - CONCURRENCY.N15.UPDATE.ALLOW_ON_SERVICE_FAILURE
  - CONCURRENCY.N15.T3501_TIMER_EXPIRY.ALLOW_ON_SERVICE_FAILURE

- CONCURRENCY.N15.TERMINATE.ALLOW_ON_SERVICE_FAILURE
- CONCURRENCY.N15.UE_NOTIFICATION.ALLOW_ON_SERVICE_FAILURE
- CONCURRENCY.N15.NOTIFICATION.ALLOW_ON_SERVICE_FAILURE
- CONCURRENCY.N15.REATTEMPT_TIMER_EXPIRY.ALLOW_ON_SERVICE_FAILURE

> **Note**
>
> If you do not make these parameters as configurable, the UE service uses the default values for these parameters.

For more information about enabling the feature through CNC Console, see PCF UE Policy Service .

- **Enable using REST API:** Set the Keys and Values for **Advanced Settings** under PCF Session Management Service. For more information about enabling the feature through REST API, see "UE Policy" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

You can configure the concurrency handling through Bulwark functionality using the CNC Console or REST API for Policy.

- **Configure using CNC Console:** Perform the feature configurations under the **Advanced Settings** section on the **PCF UE Policy Service** page. For more information about configuring audit service, see PCF UE Policy Service.
- **Configure using REST API:**Set the Keys and Values for **Advanced Settings** under PCF Session Management Service. For more information about enabling the feature through REST API, see "UE Policy" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

4.89.4 Support for Concurrency Handling using Bulwark Service in PCRF

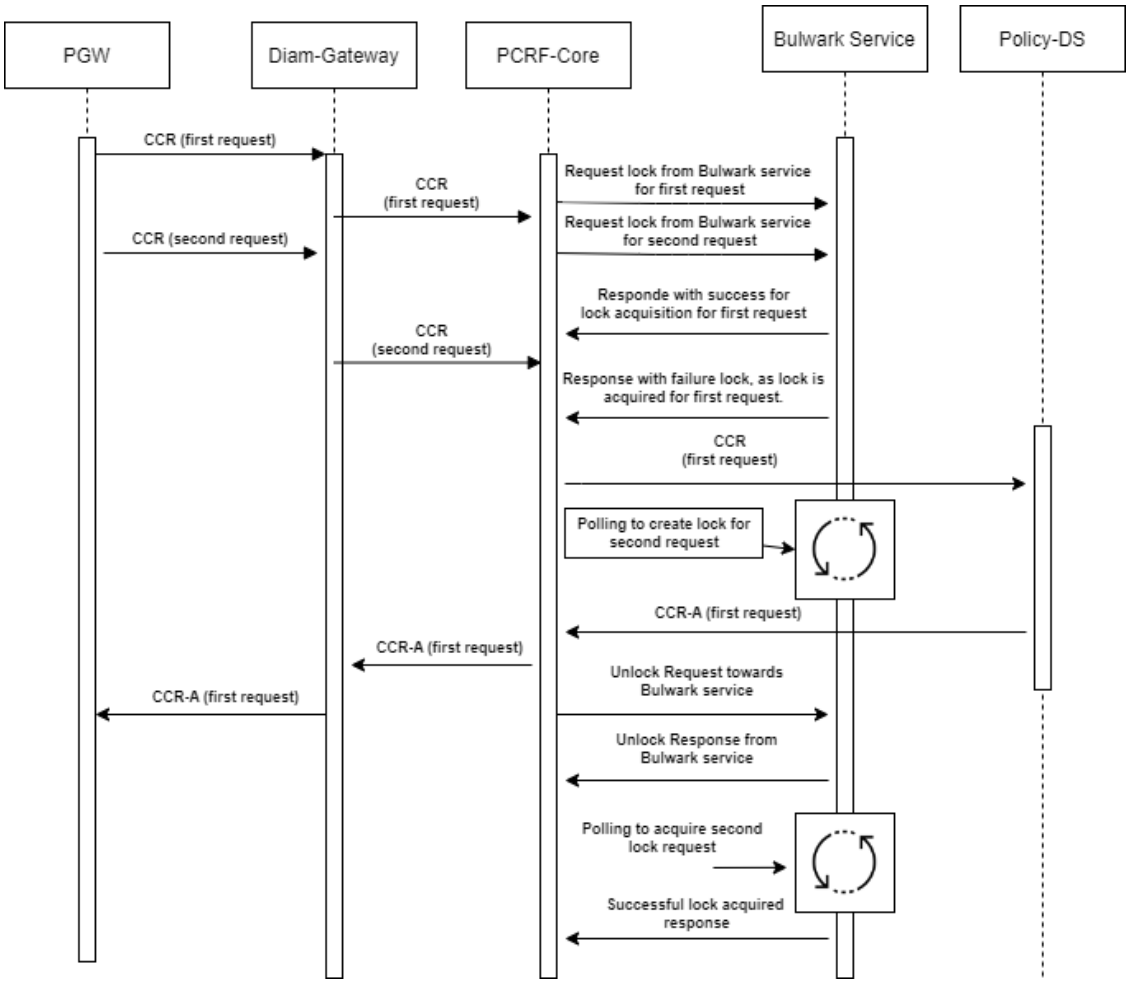
PCRF-core supports the Bulwark service to handle the concurrent messages on Gx interface. PCRF-core interacts with Bulwark service using of lock/unlock message exchanges acquiring and releasing the User-IDs based lock upon receiving CCR-I/U-T messages.

The PCRF-core sends the following parameters to request a lock:

- **Key:** The key is the identifier value with which the lock is identified. You can select either SUPI or GPSI value to be the key parameter for which the lock acquisition request must be made. By default, SUPI values are used for lock requests.
- **Lease Duration:** This value defines the duration for which lock is kept once the acquisition is successful. After this duration, the lock will be released automatically.
- **Lock Wait Timeout:** This defines the duration by which the Policy services wait for the response to get a lock. The same duration is used by Bulwark to poll for the lock in case the lock is not available. The Lock Wait timeout is also considered as the polling interval which sends the request towards Bulwark service periodically to acquire the lock for another create or delete request.
- **Retry Count:** This defines the count of retry attempts that are made by the Policy services if a lock request fails. In case all the retry attempts fail, the create or delete request is rejected. By default, 3 retry attempts are enabled.

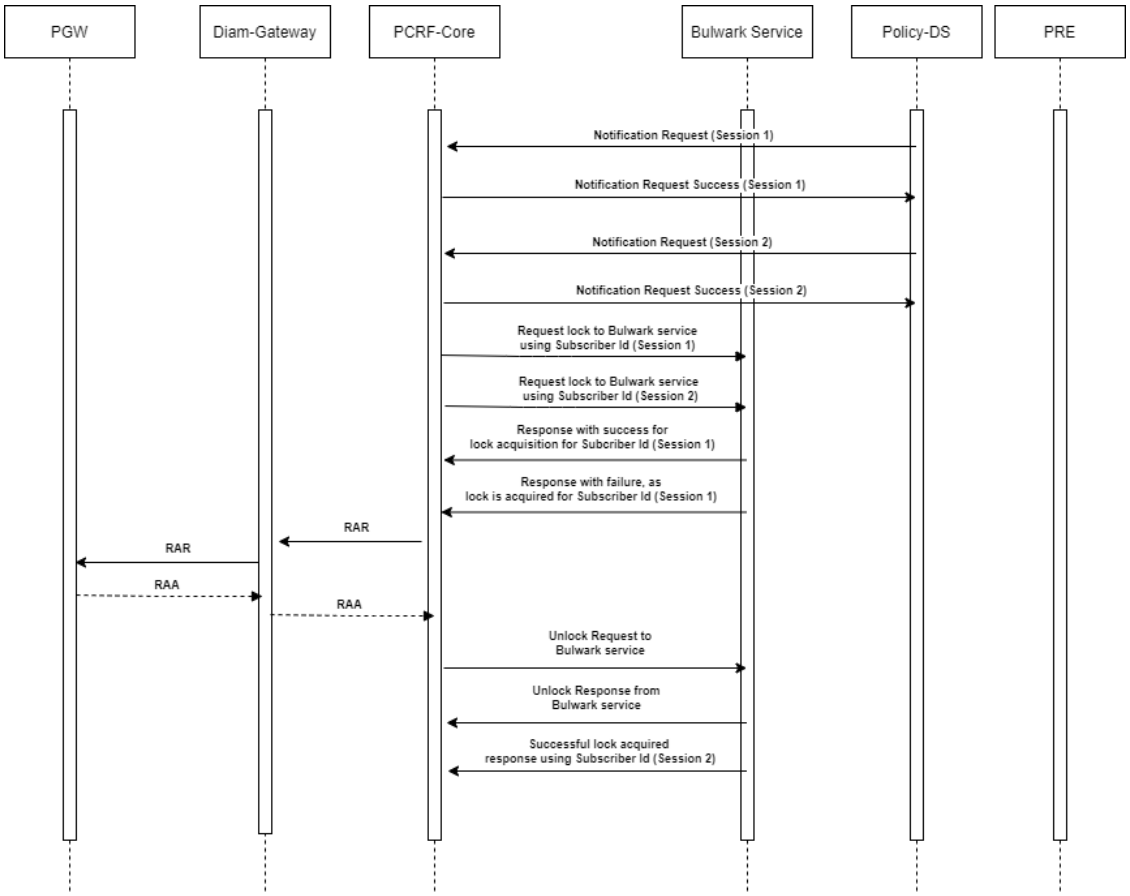
Call Flow- CCR-I, CCR-U and CCR-T messages Integration with Bulwark Service

Figure 4-138 CCR-I, CCR-U and CCR-T messages Integration with Bulwark Service



Call Flow- PDS notification (RAR) Integration with Bulwark Service

Figure 4-139 Call Flow- PDS notification (RAR) Integration with Bulwark Service



Managing Concurrency Handling for PDS Service

Enable

You can enable the concurrency handling for PCRF-Core using the CNC Console or REST API for Policy.

- **Enable using CNC Console:** Set the keys and values of the following keys under the **Advanced Settings** section on the **PCRF Core Settings** page:

Table 4-53 General

Key	Value
CONCURRENCY.ENABLED	true/false

Table 4-54 List of Keys for Gx CCR-I

Key	Default Value
CONCURRENCY.GX.CREATE.LOCK_REQUEST_RETRY_COUNT	2
CONCURRENCY.GX.CREATE.LOCK_REQUEST_RETRY_BACKOFF	1000
CONCURRENCY.GX.CREATE.LOCK_LEASE_DURATION	2000
CONCURRENCY.GX.CREATE.LOCK_WAIT_DURATION	3000
CONCURRENCY.GX.CREATE.ENABLED	True
CONCURRENCY.GX.CREATE.ALLOW_ON_SERVICE_FAILURE	True

Table 4-55 List of Keys for Gx CCR-U

Key	Default Value
CONCURRENCY.GX.MODIFY.LOCK_REQUEST_RETRY_COUNT	2
CONCURRENCY.GX.MODIFY.LOCK_REQUEST_RETRY_BACKOFF	1000
CONCURRENCY.GX.MODIFY.LOCK_LEASE_DURATION	2000
CONCURRENCY.GX.MODIFY.LOCK_WAIT_DURATION	3000
CONCURRENCY.GX.MODIFY.ENABLED	True
CONCURRENCY.GX.MODIFY.ALLOW_ON_SERVICE_FAILURE	True

Table 4-56 List of Keys for Gx CCR-T

Key	Default Value
CONCURRENCY.GX.DELETE.LOCK_REQUEST_RETRY_COUNT	2
CONCURRENCY.GX.DELETE.LOCK_REQUEST_RETRY_BACKOFF	1000
CONCURRENCY.GX.DELETE.LOCK_LEASE_DURATION	2000
CONCURRENCY.GX.DELETE.LOCK_WAIT_DURATION	3000
CONCURRENCY.GX.DELETE.ENABLED	True
CONCURRENCY.GX.DELETE.ALLOW_ON_SERVICE_FAILURE	True

Table 4-57 List of Keys for Gx RAR

Key	Defualt Value
CONCURRENCY.GX.REAUTH.LOCK_REQUEST_RETRY_COUNT	2

Table 4-57 (Cont.) List of Keys for Gx RAR

Key	Defualt Value
CONCURRENCY.GX.REAUTH.LOCK_REQUEST_RETRY_BACKOFF	1000
CONCURRENCY.GX.REAUTH.LOCK_LEASE_DURATION	2000
CONCURRENCY.GX.REAUTH.LOCK_WAIT_DURATION	3000
CONCURRENCY.GX.REAUTH.ENABLED	True
CONCURRENCY.GX.REAUTH.ALLOW_ON_SERVICE_FAILURE	True

Table 4-58 List of Keys for SD Create

Key	Default Value
CONCURRENCY.SD.CREATE.LOCK_REQUEST_RETRY_COUNT	2
CONCURRENCY.SD.CREATE.LOCK_REQUEST_RETRY_BACKOFF	1000
CONCURRENCY.SD.CREATE.LOCK_LEASE_DURATION	2000
CONCURRENCY.SD.CREATE.LOCK_WAIT_DURATION	3000
CONCURRENCY.SD.CREATE.ENABLED	True
CONCURRENCY.SD.CREATE.ALLOW_ON_SERVICE_FAILURE	True

Table 4-59 List of Keys for Rx AAR-I

Key	Default Value
CONCURRENCY.RX.CREATE.LOCK_REQUEST_RETRY_COUNT	2
CONCURRENCY.RX.CREATE.LOCK_REQUEST_RETRY_BACKOFF	1000
CONCURRENCY.RX.CREATE.LOCK_LEASE_DURATION	2000
CONCURRENCY.RX.CREATE.LOCK_WAIT_DURATION	3000
CONCURRENCY.RX.CREATE.ENABLED	true
CONCURRENCY.RX.CREATE.ALLOW_ON_SERVICE_FAILURE	true

Table 4-60 List of Keys for Rx AAR-U

Key	Default Value
CONCURRENCY.RX.MODIFY.LOCK_REQUEST_RETRY_COUNT	2
CONCURRENCY.RX.MODIFY.LOCK_REQUEST_RETRY_BACKOFF	1000

Table 4-60 (Cont.) List of Keys for Rx AAR-U

Key	Default Value
CONCURRENCY.RX.MODIFY.LOCK_LEASE_DURATION	2000
CONCURRENCY.RX.MODIFY.LOCK_WAIT_DURATION	3000
CONCURRENCY.RX.MODIFY.ENABLED	true
CONCURRENCY.RX.MODIFY.ALLOW_ON_SERVICE_FAILURE	true

Table 4-61 List of Keys for Rx STR

Key	Default Value
CONCURRENCY.RX.DELETE.LOCK_REQUEST_RETRY_COUNT	2
CONCURRENCY.RX.DELETE.LOCK_REQUEST_RETRY_BACKOFF	1000
CONCURRENCY.RX.DELETE.LOCK_LEASE_DURATION	2000
CONCURRENCY.RX.DELETE.LOCK_WAIT_DURATION	3000
CONCURRENCY.RX.DELETE.ENABLED	true
CONCURRENCY.RX.DELETE.ALLOW_ON_SERVICE_FAILURE	true

Table 4-62 List of Keys for Additional Properties

Key	Default Value
bulwark.service.enabled	true
bulwark.service.http2.enabled	true
bulwark.service.url	http://10.75.241.7:30900/v1/locks
bulwark.service.connector.timeout	3000

For more information about enabling the feature through CNC Console, see Settings.

- **Enable using REST API:** Set the Keys and Values for **Advanced Settings** under PCRF Core Settings. For more information about enabling the feature through REST API, see "Core Services" section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

You can configure the concurrency handling for PDS Service using the CNC Console or REST API for Policy.

- **Configure using CNC Console:** Perform the feature configurations under the **Advanced Settings** section on the **PCRF Core Settings** page. For more information about the configuration, see Settings.
- **Configure using REST API:** Set the Keys and Values for **Advanced Settings** under PCRF Core Settings. For more information about enabling the feature through REST API, see "Core Services" section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Observe

PCRF-Core provides the following metrics specific to concurrency handling in PCRF:

- occnp_http_bulwark_lock_request_total
- occnp_http_bulwark_lock_response_total
- occnp_http_bulwark_unlock_request_total
- occnp_http_bulwark_unlock_response_total

For more details on metrics, see PCRF Core Metrics.

4.89.5 Support for Concurrency Handling using Bulwark Service in PDS

When PDS receives concurrent notification requests for a subscriber (same SUPI or GPSI) of the same data source, processing multiple notification requests at the same time can result in inaccurate data.

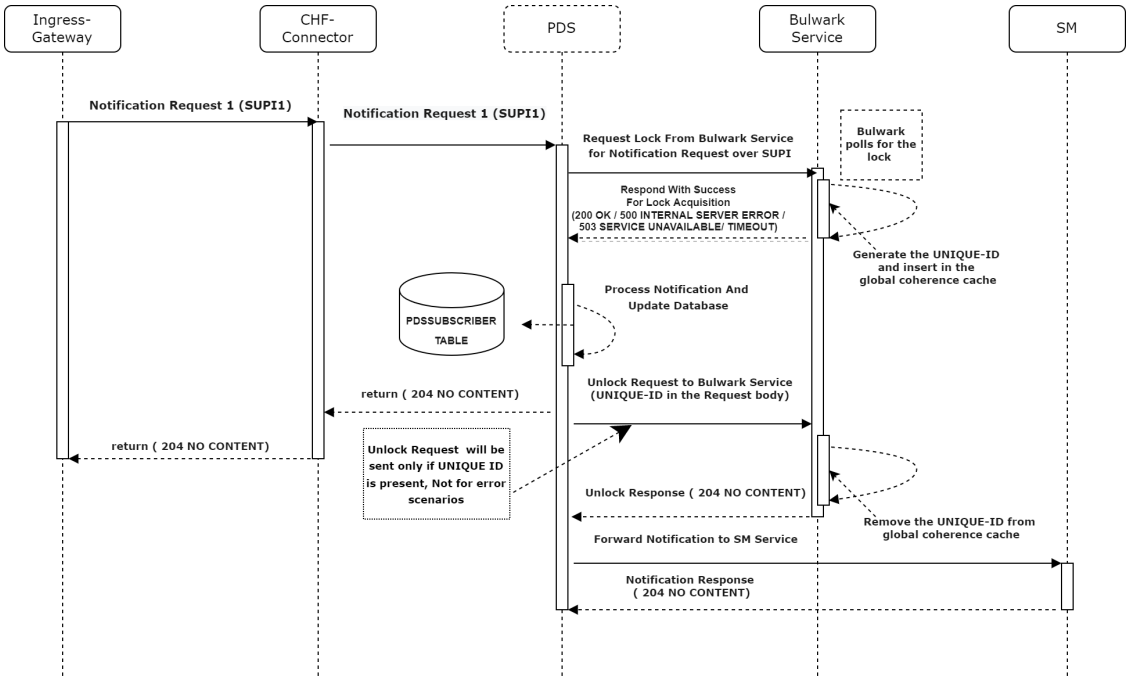
To avoid this issue, PDS is integrated with Bulwark Service. Bulwark Service provides lock and unlock mechanism over a SUPI or GPSI key and allows processing of only one notification at a time.

When PDS service receives multiple notifications from UDR, CHF or OCS, PDS requests for a lock to Bulwark service by calling the lock API service. Bulwark service approves lock for one request at a time. The other services retry to get the lock..

Once the lock is acquired, PDS service processes the notification for the SUPI or GPSI value with which the lock is requested. After the message is completed, PDS service sends a release lock request to Bulwark to release the lock.

Call Flow for Handling concurrent requests for PDS service

Figure 4-140 Call Flow - Bulwark Lock Request For CHF Notification

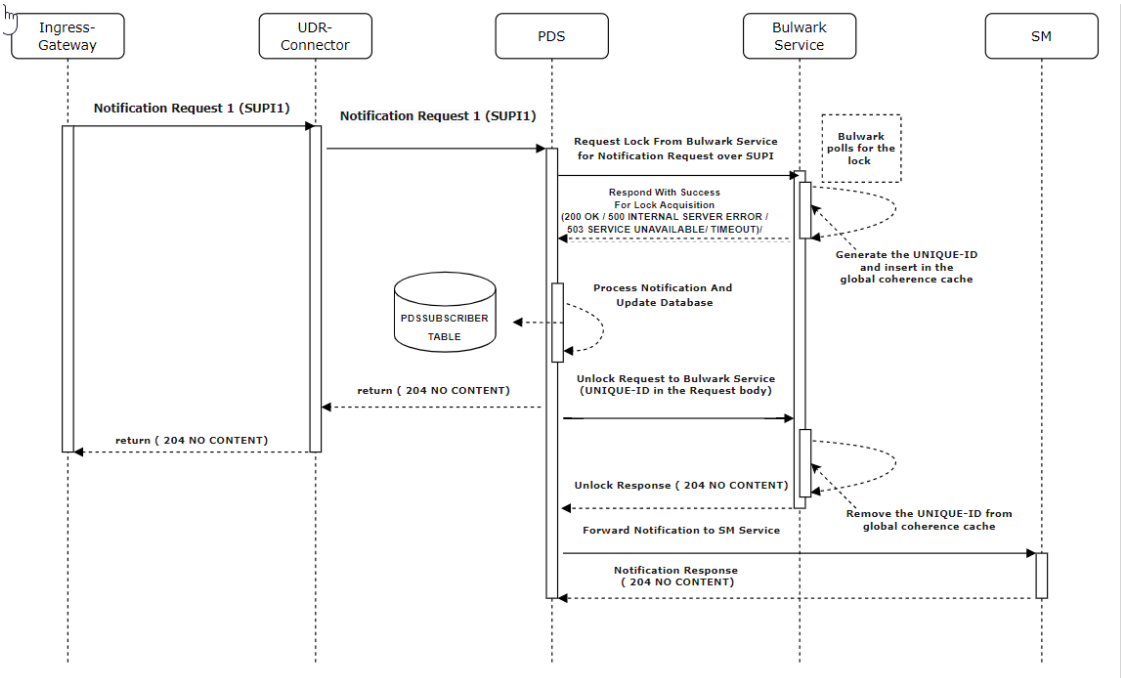


To enable the concurrency handling for CHF notifications, set the value of CONCURRENCY.BULWARK_ENABLED_FOR_CHF_NOTIFICATION to true.

In the above example, PDS receives notification request for SUPI1 from CHF through Ingress Gateway, CHF Connector.

1. PDS queries the PDS Subscriber table to fetch the SUPI or GPSI for the given sySessionID, based on CONCURRENCY.LOCK_REQUEST_KEY_TYPE_FOR_CHF_NOTIFICATION configuration in PDS Advanced Settings page.
2. PDS sends a lock request to Bulwark Service to aquire a lock for notification request over SUPI1.
3. Bulwark Service polls for the lock.
4. If the lock acquisition is successful:
    a. Bulwark Service generates a UniqueID for the lock and updates the same in the global coherence cache.
    b. Bulwark Service responds to PDS with 200 ok message.
    c. After PDS completes processing the notification request, it sends an Unlock request to Bulwark Service with the UniqueID in request parameters.
    d. Bulwark Service removes the UniqueID from the global coherence cache and responds to PDS with a 204 no content message.
    e. PDS forwards the notification request to core services and receives a response.
    f. If the lease duration expires before the processing of the notification request is complete, Bulwark Services releases the lock.
5. If the lock acquisition fails, PDS receives an error from Bulwark Service. If the same error code is configured in BULWARK_ERROR_CODES in PDS, PDS rejects the notification and responds with a 500 message to CHF connector. If the error code is not configured in BULWARK_ERROR_CODES, PDS continues to process the notification without the lock.

Figure 4-141 Call Flow - Bulwark Lock Request For UDR Notification



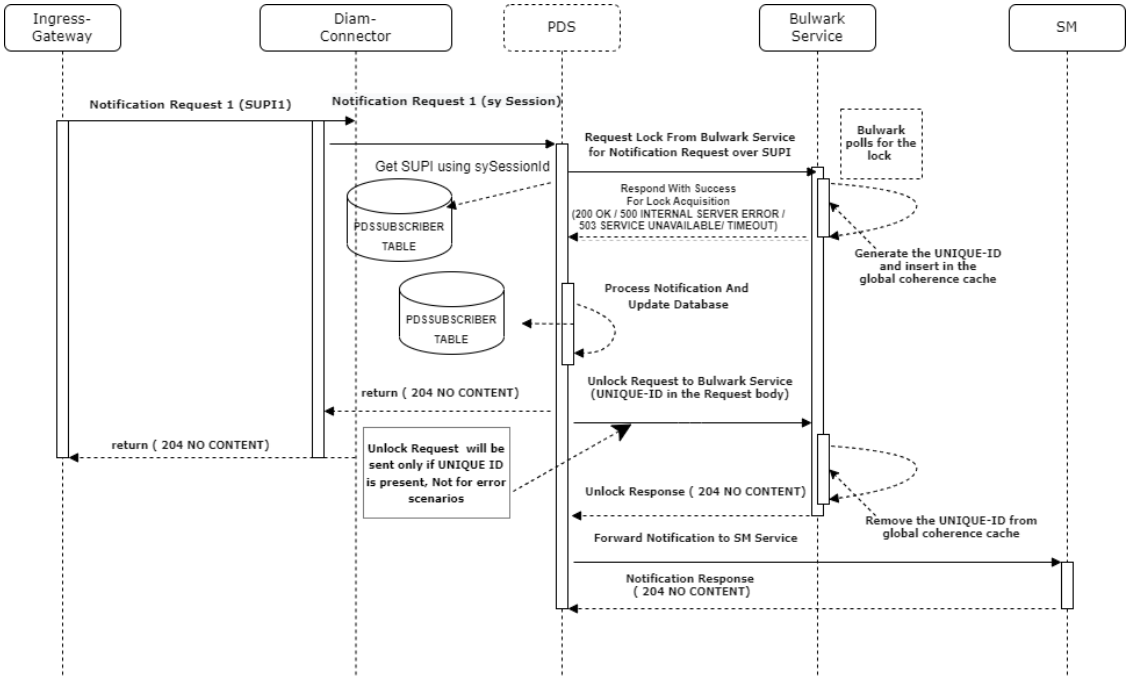
To enable the concurrency handling for UDR notifications, set the value of CONCURRENCY.BULWARK_ENABLED_FOR_UDR_NOTIFICATION to true.

In the above example, PDS receives notification request for SUPI1 from UDR through Ingress Gateway, UDR Connector.

1. PDS queries the PDS Subscriber table to fetch the SUPI or GPSI for the given sySessionID, based on CONCURRENCY.LOCK_REQUEST_KEY_TYPE_FOR_UDR_NOTIFICATION configuration in PDS Advanced Settings page.
2. PDS sends a lock request to Bulwark Service to aquire a lock for notification request over SUPI1.
3. Bulwark Service polls for the lock.
4. If the lock acquisition is successful:
    a. Bulwark Service generates a UniqueID for the lock and updates the same in the global coherence cache.
    b. Bulwark Service responds to PDS with 200 ok message.
    c. After PDS completes processing the notification request, it sends an Unlock request to Bulwark Service with the UniqueID in request parameters.
    d. Bulwark Service removes the UniqueID from the global coherence cache and responds to PDS with a 204 no content message.
    e. PDS forwards the notification request to core services and receives a response.
    f. If the lease duration expires before the processing of the notification request is complete, Bulwark Services releases the lock.
5. If the lock acquisition fails, PDS receives an error from Bulwark Service. If the same error code is configured in BULWARK_ERROR_CODES in PDS, PDS rejects the notification

and responds with a 500 message to UDR connector. If the error code is not configured in BULWARK_ERROR_CODES, PDS continues to process the notification without the lock.

Figure 4-142 Call Flow - Bulwark Lock Request For OCS Notification



To enable the concurrency handling for OCS notifications, set the value of CONCURRENCY.BULWARK_ENABLED_FOR_OCS_NOTIFICATION to true.

In the above example, PDS receives notification request for Sy Session from OCS through Ingress Gateway, Diameter Connector.

1. PDS queries the PDS Subscriber table to fetch the SUPI or GPSI for the given sySessionID, based on CONCURRENCY.LOCK_REQUEST_KEY_TYPE_FOR_OCS_NOTIFICATION configuration in PDS Advanced Settings page. .
2. PDS sends a lock request to Bulwark Service to aquire a lock for notification request over SUPI1.
3. Bulwark Service polls for the lock.
4. If the lock acquisition is successful:
   a. Bulwark Service generates a UniqueID for the lock and updates the same in the global coherence cache.
   b. Bulwark Service responds to PDS with 200 ok message.
   c. After PDS completes processing the notification request, it sends an Unlock request to Bulwark Service with the UniqueID in request parameters.
   d. Bulwark Service removes the UniqueID from the global coherence cache and responds to PDS with a 204 no content message.
   e. PDS forwards the notification request to core service and receives a response.
   f. If the lease duration expires before the processing of the notification request is complete, Bulwark Services releases the lock.

5. If the lock acquisition fails, PDS receives an error from Bulwark Service. If the same error code is configured in BULWARK_ERROR_CODES in PDS, PDS rejects the notification and responds with a 500 message to OCS. If the error code is not configured in BULWARK_ERROR_CODES, PDS continues to process the notification without the lock.

Managing Concurrency Handling for PDS Service

Enable

You can enable the concurrency handling for PDS Service using the CNC Console or REST API for Policy.

- **Enable using CNC Console:** Set value of the following keys to true under the **Advanced Settings** section on the **PDS Settings** page:
  - CONCURRENCY.BULWARK_ENABLED_FOR_CHF_NOTIFICATION
  - CONCURRENCY.BULWARK_ENABLED_FOR_OCS_NOTIFICATION
  - CONCURRENCY.BULWARK_ENABLED_FOR_UDR_NOTIFICATION

  Configuration of CONCURRENCY.BULWARK_ENABLED_FOR_CHF_NOTIFICATION, CONCURRENCY.BULWARK_ENABLED_FOR_OCS_NOTIFICATION and CONCURRENCY.BULWARK_ENABLED_FOR_UDR_NOTIFICATION are considered only if CONCURRENCY.BULWARK_ENABLED is set to true.

  CONCURRENCY.BULWARK_ENABLED is configured at the time of deployment.

  For more information about enabling the feature through CNC Console, see PDS Settings.
- **Enable using REST API:** Set the Keys and Values for **Advanced Settings** under PDS Settings. For more information about enabling the feature through REST API, see "PDS Settings" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

You can configure the concurrency handling for PDS Service using the CNC Console or REST API for Policy.

- **Configure using CNC Console:** Perform the feature configurations under the **Advanced Settings** section on the **PDS Settings** page. For more information about configuring audit service, see PDS Settings.
- **Configure using REST API:** Set the Keys and Values for **Advanced Settings** under PDS Settings. For more information about enabling the feature through REST API, see "PDS Settings" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Observe

Policy provides the following metrics specific to concurrency handling for PDS call flows:

- server_request_total
- server_response_total

For more details on metrics, see Policy DS Metrics.

4.90 Support for Sd Interface

Policy supports the Sd interface that enables it to communicate with the Traffic Detection Function (TDF). The Sd interface allows provisioning of the Application Detection and Control

(ADC) rules from Policy for traffic detection and enforcement at the TDF through Solicited Application Reporting.

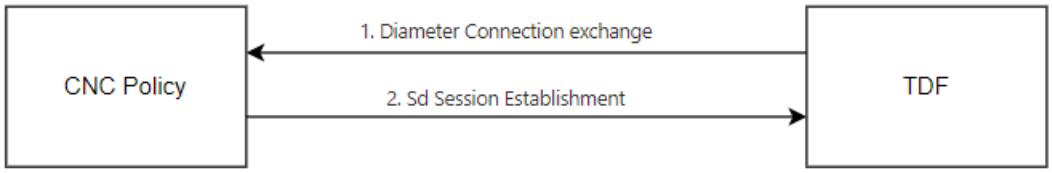
The Sd reference point exists between Policy and Charging Rules Function (PCRF) and a standalone TDF. Policy performs the following functionality for traffic detection through Sd interface:

- Establishment of session with TDF over Sd interface
- Termination of an existing TDF session
- Provisioning of ADC rules or decisions from for the purpose of traffic detection and enforcement at the TDF
- Reporting of the start and stop of detected applications and transfer of service data flow descriptions for detected applications, if available, from the TDF

The feature is supported for the Policy, PCRF, and the Converged Policy mode. It also supports the georedundant deployments of Policy.

The following diagram describes the session estblishment between Policy and TDF:

Figure 4-143 TDF Session Establishment over Sd Interface



1. Policy acts as a diameter server for CER/CEA connection exchange. It accepts the diameter CER from TDF clients.

   > **Note**
   >
   > Policy supports the diameter CER only from the configured TDF clients.

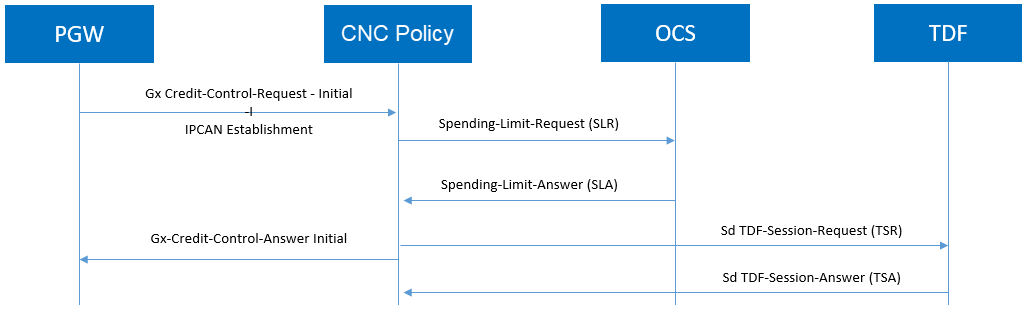
   This establishes an Sd session as a result of IP-CAN (Gx) session establishment based on Policy action.
2. Policy sends a TDF-Session-Request (TSR) to the identified TDF in order to establish the session and provide the application detection information in the ADC rules. For more information, see 3GPP Technical specification, Release 16.4, Policy and Charging Control (PCC).

Call Flow

Call Flow for TDF Session

The following diagram shows a call flow between Policy and TDF through the asynchronous mode of communication between Gx and Sd interface:

Figure 4-144 Call Flow for TDF Session



> **Note**
>
> Policy supports one Sd session per IPCAN Gx session to a RD. In case of multiple IPCAN, the new Sd Session is established to the same TDF for which the IPCAN session already exists. Multiple Sd sessions for the same Gx IPCAN session are not supported.

Managing Traffic Handling using Sd Interface

Enable

To enable the traffic handling through Sd interface functionality, Policy communicates with the OCS service. Therefore, it is required to enable the OCS Spending limit feature in PCRF-Core settings. You can enable OCS Spending limit using the CNC Console or REST API for Policy.

- **Enable using CNC Console:** Enable the OCS Spending limit functionality using the **OCS Spending Limit Attributes** on the **Settings** page for the PCRF Core service configurations. For more information about enabling the feature through CNC Console, see PCRF Core Service Configurations.
- **Enable using REST API:** Set values for the **ocsSpendingLimit.enable** parameter in the **Core Service** configuration API. For more information about enabling the feature through REST API, see "Core Service" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

You can configure the traffic handling through Sd interface functionality using the CNC Console or REST API for Policy.

- **Configure using CNC Console:** Perform the following feature configurations using the CNC Console:
  - Set value for the TDF FQDN in the **Diameter Identity** parameter under the **PGW** section on the **Network Element** page. For more information about enabling the feature through CNC Console, see PGW.
  - Set the TDF service connection using the **Identity** parameter on the **Peer Nodes** page. For more information about enabling the feature through CNC Console, see Peer Nodes.
  - Set the TDF service connection using the **type** parameter on the **Peer Node Sets** page. For more information about enabling the feature through CNC Console, see Peer Node Sets.

- **Configure using REST API:** Policy provides the following REST API for traffic handling through Sd interface configuration:
  - Set values for the **neDiameterId** parameter in the **Network for PGW** configuration API. For more information about configuring the parameter through REST API, see "Network for PGW" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.
  - Set values for the **identity** parameter in the **Peer Nodes** configuration API. For more information about configuring the parameter through REST API, see "Peer Nodes" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.
  - Set values for the **type** parameter in the **Peer Node Sets** configuration API. For more information about configuring the parameter through REST API, see "Peer Node Sets" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

After the configurations, you can use the following blockly rules to establish traffic detection sessions with selected network element identity.

Policy supports the following blockly rules to create dynamic PCC Rules for the Sd interface:

- Establish traffic detection session with selected network element identity.
  The following screen capture shows a sample for the blockly:

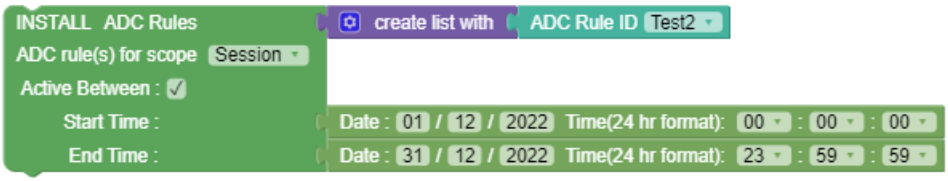
Figure 4-145 Traffic Detection for Peer Node

- Establish traffic detection session with selected Peer Node Set.
  The following screen capture shows a sample for the blockly:

Figure 4-146 Traffic Detection for Peer Node Sets

- Installation: To provision an ADC rule that has not been already provisioned. This blockly is used to install specified ADC rule(s) for selected scope active between the given start time and end time.
  The following screen capture shows a sample for the blockly:

Figure 4-147 Install with specified ADC Rules



> **Note**
>
> The Sd interface allows multiple rules to be installed with one ADC Charging Rule Install rule.

- Modification: To modify an installed ADC rule
- Removal: To remove an installed ADC rule
  The following screen captures show samples for the blockly:
  - To remove all ADC Rules

Figure 4-148 Remove all ADC Rules

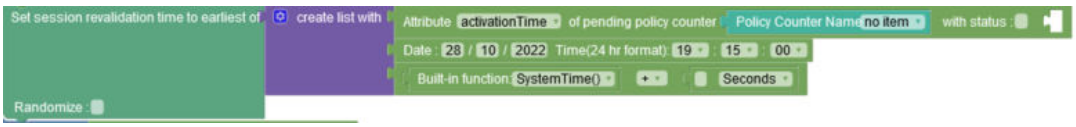
  - To remove specified ADC Rules

Figure 4-149 Remove specified ADC Rules

For more information, see "PCRF-Core Actions" in *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.

- Set Session Revalidation Time

Figure 4-150 Set Session Revalidation Time



Observe

Policy provides metrics specific to determine the successful TSR and RAR messages sent to TDF.

Policy provides following alert for this feature.

- **TDF_CONNECTION_DOWN**

For more information, see TDF_CONNECTION_DOWN.

4.91 Support for Retrying Binding Registration on BSF and generating Alarm to Operator

The Retry Binding Registration on BSF feature enhances the binding registration functionality between SM Service and BSF. SM Service sends the binding registration requests to BSF through the Binding service. To manage the registration failure scenarios, Policy provides the

Session Retry functionality. The Policy Binding service handles the Session Retry functionality and returns any of the following responses to the SM service:

- **Created without BSF Metadata:** The binding creation is successful in BSF
- **Failed with BSF Metadata:** The binding creation on BSF failed and the BSF metadata contains the problemDetails of the failure. If the error code in the BsfMeatadata does not match with the configured error code, it sends the termination notification in case the **Abort or Terminate** switch is enabled in SM configurations.
- **500 Internal Server Error:** An error occurred at the Binding service. It sends a termination notification if the **Abort or Terminate** flag is enabled in SM configurations

Starting with release 22.3.0, the Retry Binding Registration on BSF feature enhances the existing session retry mechanism by supporting multiple cycles of retry binding creation until a maximum number of attempts are reached or the response error code does not match with any of the configured error codes.

The following steps describes how PCF reattempts a binding request after receiving the failure response from BSF:

1. PCF sends a binding request to BSF and the initial request get failed.
2. PCF uses the Alternate Route Retry (ARR) functionality and sends the BSF registration requests to alternate BSFs.
3. In case all the session retry attempts configured in Policy fails, then Binding service returns to the SM service with the status code of the last binding request. The SM service evaluates if a recreate attempt must be triggered based on the following configurations:
   a. Recreate attempt feature is enabled
   b. Status code returned by Binding service matches with one of the error codes configured in the retry profile for BSF.

      > **Note**
      >
      > If no error codes have been configured in the retry profile for BSF, no recreate attempt is triggered.

   c. The maximum number of recreate attempts has not been reached. Based on the configurations, the SM service triggers the binding request after the duration configured through the BOT configuration is over.

      > **Note**
      >
      > The binding operation must be in asynchronous mode. For more information about configuration, see Managing Retrying Binding Registration on BSF.
4. If all the configurations are matched, then the SM service initiates a recreate attempt.
5. In case of the failure of the recreate attempt, the SM service reattempts the requests for the number of times configured in the **Maximum Number of Attempts** parameter in the **Reattemts Profile**.
6. If all the recreate attempts get failed and the **Abort or Terminate Session on Binding Error** parameter value under the PCF Session Management configuration is set to **True**, then PCF sends the termination notification to the SMF. In case the parameter value is set to **False**, PCF does not terminate the SM session. For more information about configuration, see Managing Retrying Binding Registration on BSF.

Handling of Pending Operations by SM Service

The SM service maintains a record of the pending binding requests in the Pending Operations database table. For any binding create request failure, if the error response matches with the configured error codes, then SM service creates a record in the Pending Operations database table with the key details. The Pending Operations table can be configured using the **Audit** configurations of the **PCF Session Management** configurations.

Handling of Pending Operations by Audit Service

The Policy Audit service controls the notification flow for binding registrations. If enabled, the Audit service records the pending operation details in the Policy database and reports it to the SM Service in case of BSF registration failure reaches the threshold value configured by the operator.

In case of registration failure with multiple BSF instances, the database entries increases and causes congestion in the SM service while trying to process all of the pending binding requests. To handle such scenarios, SM service continuously calls the Audit Service API to check on the number of records in the table and validate if it has crossed the threshold value. On receiving the notification, the SM service rejects all the SM Create requests with a DNN for which binding has to be setup. The rejection of requests is stopped once the *PendingOperation* table records reaches below the set threshold.

For more information on configuring the *PendingOperation* settings, see Audit Service Configurations in the Managing Retrying Binding Registration on BSF .

Site Takeover

The *PendingOperation* table contains the details of the site that is used to process the failed SM Create requests for PCF Binding. Each site processes the entries that belong to its site. In case a site goes down and the operator wants another site to process the pending operations, the Site Takeover functionality can be enabled. For this it is necessary to create a site takeover profile specifying the site id of the site that is failing. For more information on configuring the alternate site for takeover, use the **Site Takeover** Configurations on the CNC Console for Policy.

Managing Retrying Binding Registration on BSF

Enable and Configure

You can enable and configure Retrying Binding Registration on BSF using the CNC Console or REST API for Policy.

- **Configure using CNC Console:** Perform the following feature configurations using the CNC Console:
  - Configure the Reattempt profile for BSF retry registration requests using the **Reattempts Profile** page. For more information, see Reattempts Profile.
  - Configure the error codes for which the reattempt must be initiated using the **Attempt Alternate Route for Following Requests** under the **Alternate Routing Settings** of the **Retry Profile** page. For more information, see Retry Profiles.
  - Enable the Retry BSF Registration functionality using the **Binding Configuration** section on the **PCF Session Management** page. For more information, see PCF Session Management.
  - Configure the Pending Operation table using the **Audit** section on the **PCF Session Management** page. For more information, see PCF Session Management.
  - Configure an alternate site that can take over the current site to manage the pending binding records in case of failure using the **Site Takeover** page. For more information, see Site Takeover.

  - You can use the **Set Binding Registration to** blockly rule to indicate if Binding Registration is required for managing the pending operation table of the Audit service.
  - To configure the frequency with which SM service calls the Audit service API for checking the Pending Operation table threshold, use the **Advance Settings** section on **PCF Session Management** page. For more information, see PCF Session Management.

- **Configure using REST API:** Perform the following feature configurations using the REST API Interface for Policy:
  - Configure the Reattempt profile for BSF retry registration requests using the **Reattempts Profile** configuration API.
  - Configure the error codes for which the reattempt must be initiated using the **alternateRoutingSettings.errorCodesList** parameter in the **Retry Profile** configuration API.
  - Enable the Retry BSF Registration functionality using the **BindingConfiguration** parameters of the **Session Management Service** configuration API.
  - Configure the Pending Operation table using the **Audit** parameters of the **Session Management Service** configuration API.
  - Configure an alternate site that can take over the current site to manage the pending binding records in case of failure using the **siteTakeover** configuration API.
    For more information about configuring the parameters through REST API, see *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Observe

Policy uses the following metrics to observe the Retrying Binding Registration on BSF functionality:

- occnp_pending_binding_reattempts_total
- occnp_pending_binding_terminate_all_attempts_failed_total
- occnp_pending_binding_reattempt_fail_total
- occnp_pending_operation_threshold_reached_total
- occnp_reject_sm_create_threshold_reached_total
- occnp_audit_db_records_count_belonging_to_site

Policy uses the following alerts to monitor the pending operations:

- pcf_pending_binding_records_count
- pcf_pending_binding_site_takeover
- pcf_pending_binding_threshold_limit_reached

4.92 Controlled Shutdown of an Instance

Policy supports controlled shutdown feature to provide the partial or complete isolation of the site from the network so that the operator can perform necessary recovery procedures when required. It helps the operator to perform the recovery procedures as per the requirement.

The site isolation is achieved by shutting down the load at gateways (Ingress Gateway, Egress Gateway, and Diameter Gateway) and updating the NF status as SUSPENDED at NRF.

> **Note**
>
> Ingress Gateway Service must be enabled for the working of this feature.

Operational State

The site can be in one of the three possible operational states NORMAL, PARTIAL SHUTDOWN, or COMPLETE SHUTDOWN. The operational state can move to any of the states from the current state, there is no definitive order of state change. Currently, the operational state is stored in the common config server of the Ingress Gateway. It is read by Ingress Gateway, Egress Gateway, Diameter Gateway, and App-info periodically and action is triggered based on the current state.

> **Note**
>
> Since the operational state is stored in config server, the service instances will detect the state change after the config refresh is done. If the config refresh interval is set as 5 seconds, then the pods may recognize the operational state change after 5 seconds.

The operational state can be modified through CNC Console or REST API. Operation state configuration stored in the common config server will be read by the following services:

- Ingress Gateway
- Egress Gateway
- Diameter Gateway
- App-info

> **Note**
>
> If the Disaster Recovery procedure is performed when the config backup was taken when the system was in PARTIAL or COMPLETE SHUTDOWN state, then manual intervention may be required to change the operational state back to NORMAL state.

Load Control

Gateways enforce load control when the system is in a PARTIAL or COMPLETE SHUTDOWN state. The level of load control varies based on the shutdown state. When in a PARTIAL SHUTDOWN state, no new session establishments are allowed so session creation messages will be rejected (with configured error code) in this state. When in complete shutdown, no messages are allowed.

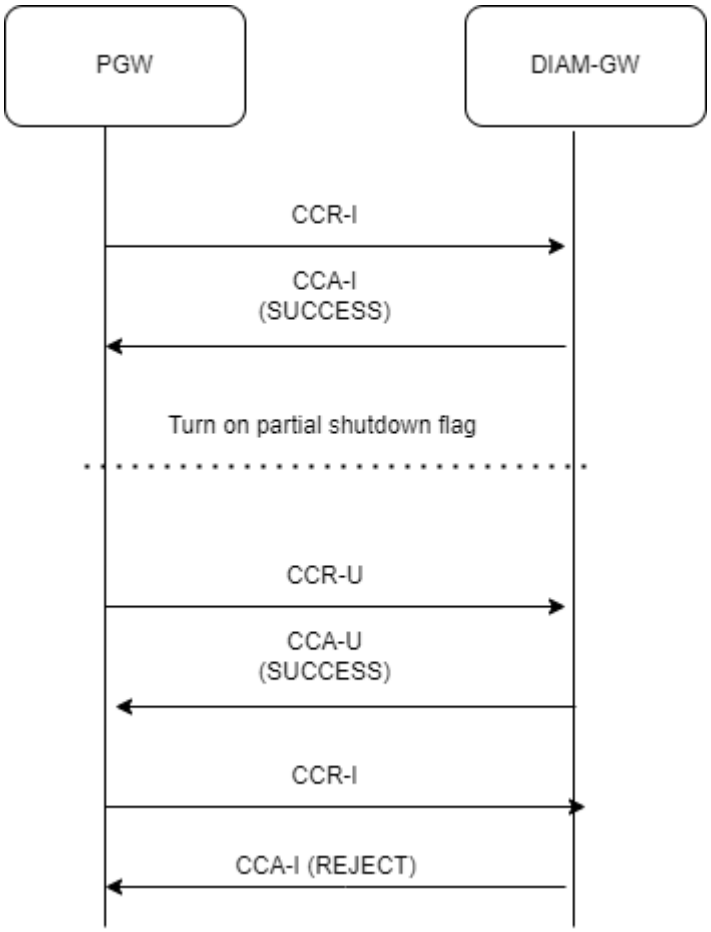
> **Note**
>
> When the system is in COMPLETE SHUTDOWN state, audit service triggered notification or diameter messages will be rejected at respective gateways.

Call Flow for Diameter Gateway

NORMAL State: If the Controlled shutdown operational state is NORMAL, then the Diameter Gateway processes the message as normal.

PARTIAL SHUTDOWN: If the controlled shutdown operational state is PARTIAL SHUTDOWN, then the Diameter Gateway accepts only in-session messages and rejects all CCR-I and AAR-I messages.

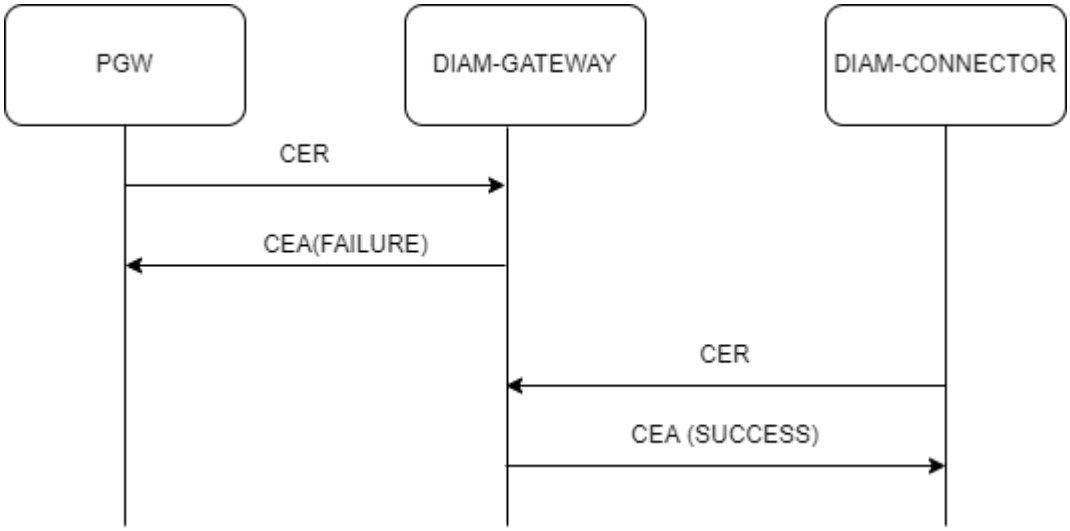
Figure 4-151 Call Flow for Diameter Gateway PARTIAL SHUTDOWN



When the operational mode is NORMAL, CCR-I is success. When operational mode is changed to PARTIAL SHUTDOWN, CCR-U is success but CCR-I is rejected.

COMPLETE SHUTDOWN: If the controlled shutdown operational state is COMPLETE SHUTDOWN, then the Diameter Gateway accepts only in-session messages and rejects all messages.

Figure 4-152 Call Flow for Diameter Gateway COMPLETE SHUTDOWN



When the operational mode is COMPLETE SHUTDOWN, CER is rejected when coming from external NF, but is successful when coming from backend.

Call Flow for Egress Gateway

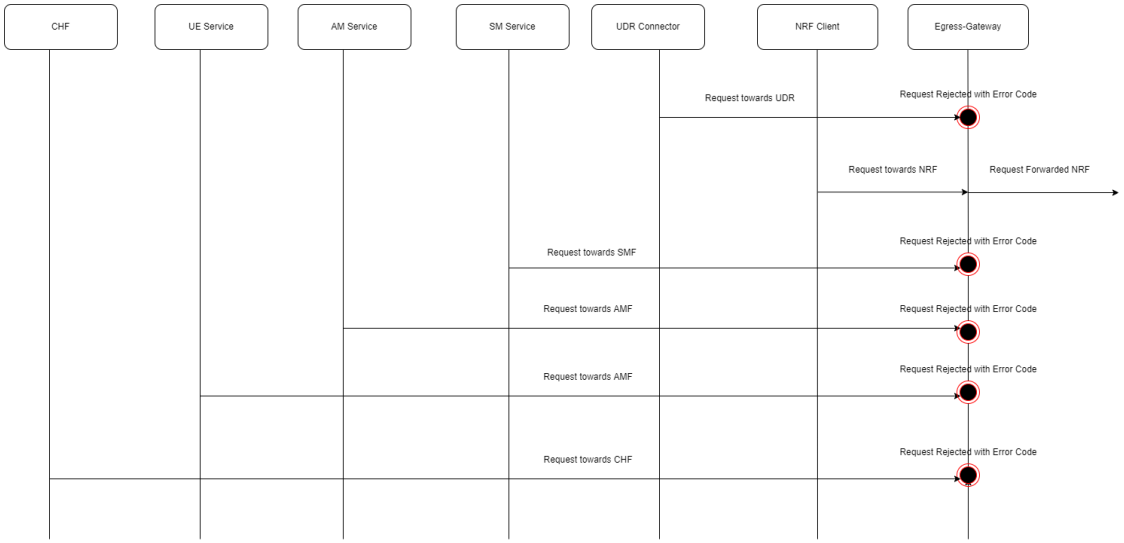
NORMAL State: If the Controlled shutdown operational state is NORMAL, then the Egress Gateway processes the message as normal.

PARTIAL SHUTDOWN: If the controlled shutdown operational state is PARTIAL SHUTDOWN, then the Egress Gateway processes the message as normal.

COMPLETE SHUTDOWN: If the controlled shutdown operational state is COMPLETE SHUTDOWN, then the Egress Gateway processes the request as follows:

- Forward all requests received from NRF Client.
- Reject all requests received from any other services like UDR Connector, SM Service, AM Service, UE Service, and CHF Connector.

Figure 4-153 Call Flow for Egress Gateway COMPLETE SHUTDOWN



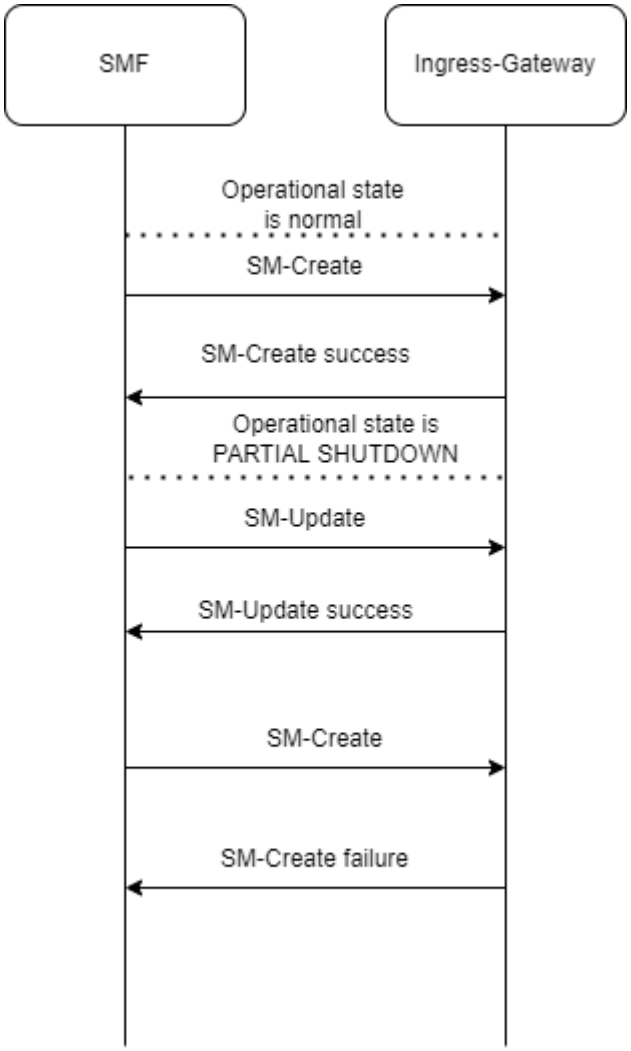
SM-Policy data request from UDR connector towards Egress Gateway will not be processed in case of COMPLETE SHUTDOWN.

Call Flow for Ingress Gateway

NORMAL State: If the Controlled shutdown operational state is NORMAL, then the Ingress Gateway processes the message as normal.

PARTIAL SHUTDOWN: If the controlled shutdown operational state is PARTIAL SHUTDOWN, then the Ingress Gateway accepts only in-session messages and rejects all SM-Create requests.

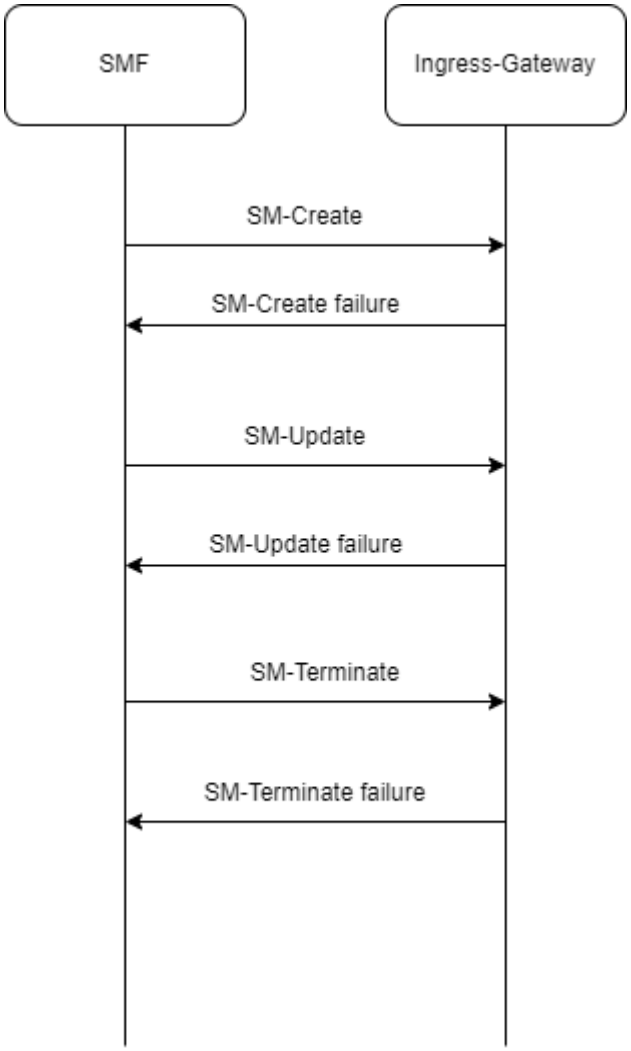
Figure 4-154 Call Flow for Ingress Gateway PARTIAL SHUTDOWN



When the operational mode is NORMAL, SM-Create is success. When operational mode is changed to PARTIAL SHUTDOWN, SM-Update is success but SM-Create is rejected.

COMPLETE SHUTDOWN: If the controlled shutdown operational state is COMPLETE SHUTDOWN, then the Ingress Gateway rejects all incoming requests.

Figure 4-155 Call Flow for Ingress Gateway COMPLETE SHUTDOWN



In Ingress Gateway COMPLETE SHUTDOWN, all the requests coming from external NFs are rejected.

App-info

App-info calculates the service status of the NF periodically, and this calculation has three possible outcomes viz Running, Not Running, and Deregister. Whenever the system's operation state is changed to PARTIAL or COMPLETE SHUTDOWN and the calculated status is "Running" then the status will be overridden and changed to "Not Running".

Managing Controlled Shutdown of an instance

Enable

You can enable or disable the Controlled Shutdown feature by using the `enableControlledShutdown` parameter in the custom.yaml file. This parameter is set as `false` by default. You can enable it by setting its value as `true`. For more information, see *Controlled Shutdown Configurations* section in the *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, Fault Recovery Guide*.

Configure

Diameter Gateway and Ingress Gateway can be configured through CNC Console. For more information, see Controlled Shutdown Configurations.

Egress Gateway routes configuration for controlled shutdown is done through Helm. For more information, see the *Controlled Shutdown Configurations* section in the *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, Fault Recovery Guide*.

Observe

Metrics

Policy provides the following metrics specific to controlled shutdown feature:

- system_operational_state
  For more information, see CM Service Metrics.
- diam_controlled_shutdown_message_reject_total
  For more information, see Diameter Gateway Metrics.

Alerts

Policy provides the following alerts for controlled shutdown feature:

- SYSTEM_IMPAIRMENT_MAJOR
- SYSTEM_IMPAIRMENT_CRITICAL
- SYSTEM_OPERATIONAL_STATE_NORMAL
- SYSTEM_OPERATIONAL_STATE_PARTIAL_SHUTDOWN
- SYSTEM_OPERATIONAL_STATE_COMPLETE_SHUTDOWN

For more information, see Common Alerts.

4.93 Enhancement to use Cached NRF Discovery Responses

Policy communicates with different Network Functions (NFs) through the NRF-Client discovery service. As a Consumer NF, Policy registers with NRF and discovers the required Producer NF through the NRF Client discovery feature.

PCF is enabled to maintain a cache of the NF Profiles discovered autonomously through the NRF Client management. The cache stores the NF Profiles with the NF Type and locality details of the discovered NFs.

> **Note**
>
> Policy supports the enhancement to use Cached NRF discovery responses in case of failure and unusable producers for autonomous discovery only.

PCF is enabled to cache the discovered NF Profiles from NRF for autonomous discovery for producer NFs, such as CHF and BSF. It uses the cache to perform the following functions:

- Rediscovers the NF Profiles for which the validity time is expired
- Rediscovers the NF Profiles when all the discovered profiles for a particular query parameter are in the **Suspended** state, irrespective of their validity time. **Note:** PCF does not use the NFs in the suspended state.

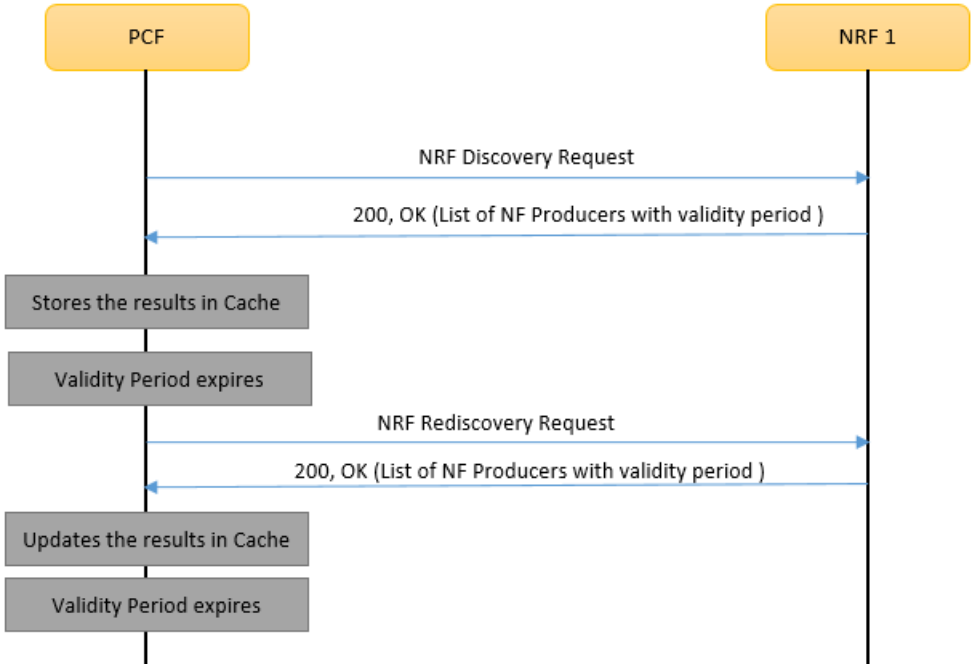
- Uses the expired discovered profiles, which are in the **Registered** state in case the NRF is unreachable during rediscovery

NF Discovery and Rediscovery Based using PCF Cache

The following call flow describes an example of PCF performing the NF discovery through NRF Client and maintaining a cache:

The call flow is described as follows:

Figure 4-156 Call Flow - NRF Rediscovery



1. During PCF deployment, operators provide a list of NFTypes, which must be automatically subscribed by the NRF Client. NRF triggers the NRF discovery operation for the set of NfTypes provided for auto-subscription.
2. NRF sends the discovery response to the PCF with the query parameters. At runtime, when there is any change in the profiles, the NRF notifies PCF about the changed profiles.
3. PCF stores the producer NF details and their query parameters in a cache. It uses the validity period parameter to calculate the expiry time the producer NF profile. Once the validity period is over, the NF profile is considered **Expired** and PCF rediscovers the NF profiles. The expiry duration can be any of the following values:
   - The validity period included in the discovery response from the NRF Client
   - The value configured in the **validityTime** parameter through the Policy Custom Values YAML file.

   Whichever is the minimum value for the validity period is considered the expiry duration.

   Example:

   During PCF deployment, the **nrf-client.profile.validityTime** parameter value is configured as 60 seconds. And the validityPeriod provided by NRF during discovery is as follows:

- For UDR: 30 seconds
- For CHF: 150 seconds
- For BSF: 90 seconds

Then, the expiry duration for the respective producer NFs is as follows:

- For UDR: 30 seconds
- For CHF: 60seconds
- For BSF: 60 seconds

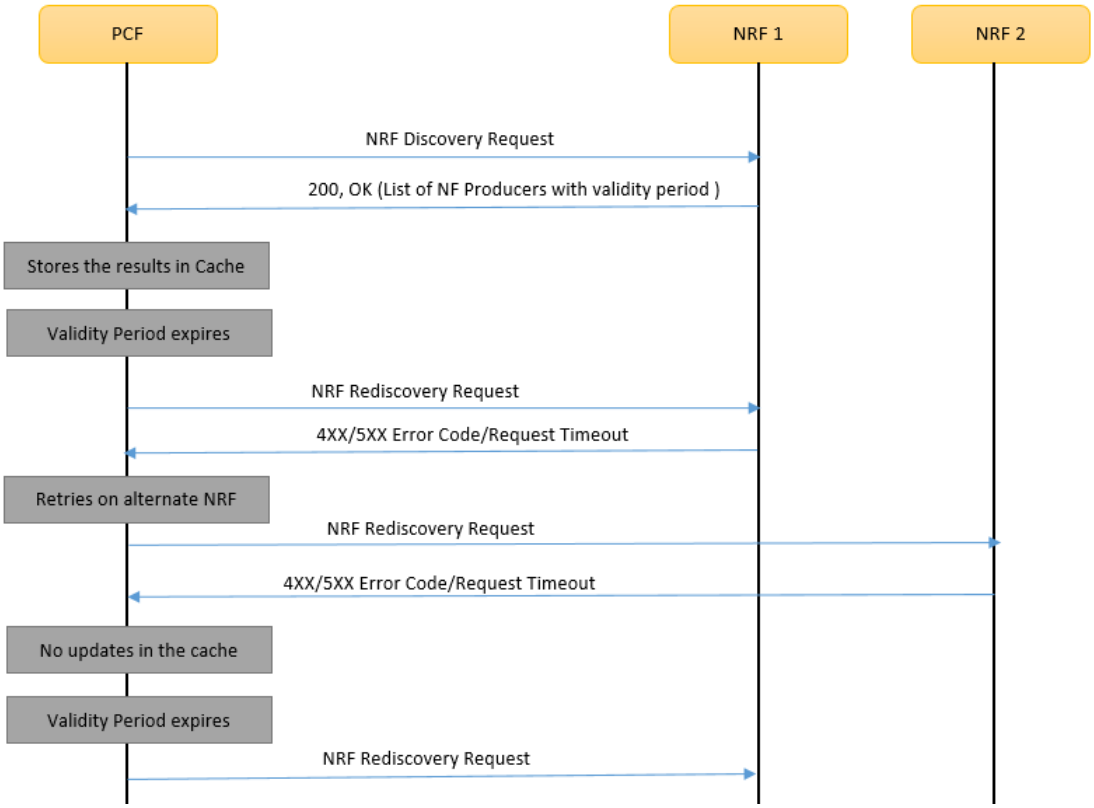
4. In case, the validity period for a producer NF expires, then PCF sends the rediscovery request to the NRF and updates the cache with the latest response.

> **Note**
>
> In case of an empty response from NRF, PCF considers it as valid response and stores in the cache for both for initial discovery and rediscovery. It sends a rediscovery request once the empty response expires.

NF Discovery Error Scenario

The call flow is described as follows:

Figure 4-157 Call Flow - NF Discovery Error Scenario



1. In the case of an Expired NF producer, PCF sends a rediscovery request to NRF.
2. PCF may attempt to send an NRF discovery request to another NRF. If another NRF still responds with an error or still timeout, PCF continues using the cache for the expired producer and after the validity time, it sends another NRF discovery request.
3. If the NRF rediscovery fails and NRF sends an error response to PCF, then the PCF keeps on using the expired records available in the cache.
4. PCF continues sending the rediscovery requests to NRF based on the expiry duration for the producer NF.

> **Note**
>
> If all the NF producers available in the PCF cache are suspended, then PCF rejects the NRF discovery request. When the cache is not yet expired, and there is no available NFs in the cache due to all of the cached NFs having gone into suspend status, ignore the TTL & trigger a discovery request to the NRF.

Managing the Enhancement to use Cached NRF Discovery Responses

Enable

To enable the PCF cache for NRF Discovery results set value of the **nrf-client.profile.cacheDiscoveryResults parameter** to **true** using the Custom Values YAML file for Policy.

Configure

You need to configure the Cahe of NRF Discovery Result functionality while deploying the PCF. The following parameters must be updated in the Custom Values YAML file for Policy:

Table 4-63 NRF Rediscovery Paramaters

Parameter	Description
discoveryRefreshInterval	This attribute defines the maximum ValidityPeriod at which the discovery results shall be refreshed. The ValidityPeriod received in the discovery response shall be capped at this value. If ValidityPeriod received in discovery results is 60s. The validityPeriod shall be capped to 10s as per configuration. If ValidityPeriodn received in discovery results is 5s. No capping shall be applied and valdiityPeriod shall be considered as 5s. Default: 10
discoveryDurationBeforeExpiry	This attributes defines the rate at which the NF shall resend discovery requests to NRF. The value shall be configured in terms of percentage(1-100). if the discovery ValidityPeriod is 10s(after applying the capped value of discoveryRefreshInterval), then the discovery requests shall be sent at discoveryDurationBeforeExpiry * 10/100.

Table 4-63 (Cont.) NRF Rediscovery Paramaters

Parameter	Description
enableDiscoveryRefresh	Feature flag to enable Automatic Discovery Refresh. Default: false
enableRediscoveryIfNoProdNFs	Feature flag to enable automatic rediscovery in case no producer NF is available. Default: false
offStatesForRediscoveryIfNoProdNFs	The NF producer status for which discovery request must be sent to NRF. It indicates the nfStatus to be considered for unavailability of all profiles and performing Rediscovery: Provide comma separated value for desired states. Possible values: • SUSPENDED • UNDISCOVERABLE • DEREGISTERED
discoveryRetryInterval	The Retry Interval after a failed autonomous discovery request.

4.94 Limiting the Number of Sessions

Limiting the Number of Sessions feature is implemented to manage and control the usage of resources. When this feature is enabled, Policy allows you to configure the maximum number of active sessions. It avoids multiple policy association table entries for a subscriber, in the respective policy service database tables (such as SMPolicyAssociations, AMPolicyAssociations, and UEPolicyAssociations)

Limiting the number of sessions for AM service and UE Policy service

The following are the parameters used to enable maximum sessions limit for each of the services:

- **Enable Max Session Limit** parameter for AM Service
- **Enable Max Session Limit** parameter for UE Policy Service

When the Policy service receives request for a new session and if the maximum sessions limit for the subscriber is enabled, it checks if there are existing sessions for the same subscriber (SUPI).

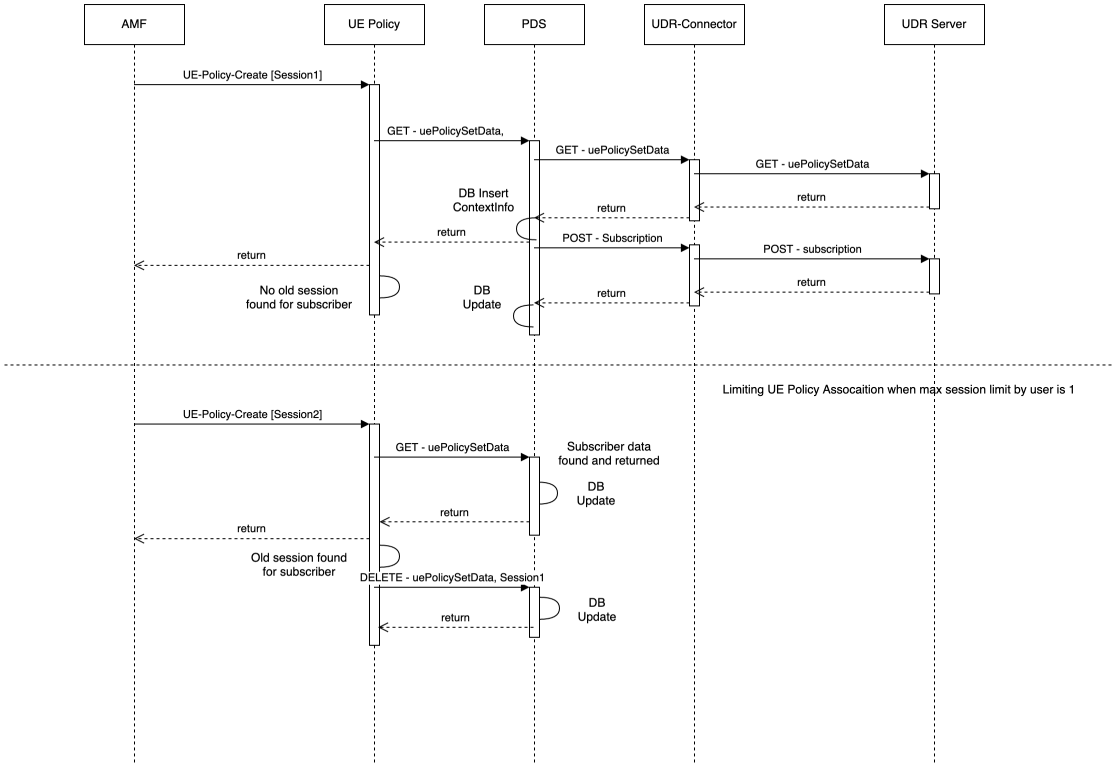
If the number of sessions for the same subscriber (SUPI) exceeds the **Max Session Limit Per User** configured, it creates a policy association for the new session in its respective database and deletes the oldest session based on the session created timestamp. The policy service initiates a local delete request.

> **Note**
>
> In Kubernetes deployment resource AM Service and UE Policy Service, the `LIMITING_SESSION_DELETE_DELAY_MILLIS` environment variable is configured to set a delay for initiating the local delete of the older session when the number of sessions for a subscriber (SUPI) exceeds the Max Session Limit Per User. The default value of `LIMITING_SESSION_DELETE_DELAY_MILLIS` is 3000 milliseconds.

For example, if the **Max Session Limit Per User** configured for UE Policy Service is 2. When a third session is created for a subscriber with two existing sessions, the UE Policy Service creates a UEPolicyAssociation for the new session and deletes the oldest session from UE Policy database. The policy service initiates a local delete request.

> **Note**
>
> The oldest session is deleted only from the local policy service and PolicyDS databases if the number of sessions exceeds the **Max Session Limit Per User**.

Figure 4-158 Limiting the number of sessions for UE Policy Service



Limiting the number of sessions per subscriber and DNN+SNSSAI for SM service and PCRF Core service

Whenever there is a CREATE request for a new SM session or PCRF Core session, Policy queries the database using SUPI or GPSI or using both SUPI and GPSI. If this feature is enabled, there are two levels at which the number of sessions are limited:

Limiting the maximum number of bindings at DNN + SNSSAI level

This is the first level of limiting the bindings at Binding service.

When limiting the maximum number of bindings at DNN + SNSSAI, Binding service limits the bindings based on the context owner (SM service/ PCRF Core service).

> **Note**
>
> SNSSAI is considered depending on the Binding configuration in CNC Console.

You can enable and configure limiting the number of bindings at DNN + SNSSAI level by configuring **Max Session Limit By APN** field in **Binding Service** page under **Service Configurations** in CNC Console.

The default limit is 2 per DNN+SNSSAI+IP domain combination.

Limiting the maximum number of bindings at user level

This is the second level of limiting the bindings at Binding service.

On top of the first level limit, user level limiting is applied. Based on the timestamp, sessions older than the configured limit, (irrespective of the DNN+SNSSAI limit) are marked as stale and are cleanedup.

You can enable limiting the number of bindings at user level using **Session Limit By User** field in **Binding Service** page under **Service Configurations** in CNC Console.

You can configure the number of bindings at user level by configuring **Max Session Limit By User** field in **Binding Service** page under **Service Configurations** in CNC Console.

The default limit is 10 per user level.

> **Note**
>
> Binding service identifies the sessions as stale and it notifies SM service or PCRF Core service based on the context owner to perform the clean up.

Example 1: If Binding session Limit at DNN+SNSSAI level is configured as 2 for internet sessions, after the first two sessions are created for the user, if Binding service receives request for the creation of 3rd session, it creates the 3rd session and marks the 1st session as stale.

Table 4-64 Binding Session Limit at DNN+SNSSAI level -2

Session ID	Context Owner	DNN+SNSSAI	Timestamp	isStale	Comments
Session1	PCF-SM	Internet	'2021-05-05 01:00:00'	0	First session creation for internet session
Session2	PCF-SM	Internet	'2021-05-05 02:00:00'	0	Second session creation for internet session
Session3	PCF-SM	Internet	'2021-05-05 03:00:00'	0	Third session creation for internet session At this point DNN+SNSSAI limit exceeds. Binding service marks Session1 as stale by changing the value of isStale to 1. It indicates SM service to cleanup Session1.

Example 2: If the Binding session limit for the user is configured at level 5, after creating the first five sessions for the user, if Binding service receives request for 6th session, it creates the 6th session and marks the 1st session as stale.

Table 4-65 Binding Session Limit at User level - 5

Session ID	Context Owner	DNN+SNSSAI	Timestamp	isStale	Comments
Session1	PCF-SM	Internet	'2021-05-05 01:00:00'	0	First session creation for internet session
Session2	PCF-SM	Internet	'2021-05-05 02:00:00'	0	Second session creation for internet session
Session3	PCF-SM	ims	'2021-05-05 03:00:00'	0	First session creation for IMS session
Session4	PCF-SM	ims	'2021-05-05 04:00:00'	0	Second session creation for IMS session
Session5	PCF-SM	gaming	'2021-05-05 05:00:00'	0	First session creation for gaming session

Table 4-65 (Cont.) Binding Session Limit at User level - 5

Session ID	Context Owner	DNN+SNSSAI	Timestamp	isStale	Comments
Session6	PCF-SM	gaming	'2021-05-05 04:00:00'	0	Second session creation for gaming session At this point user limit exceeds. Binding service marks Session1 as stale by changing the value of isStale to 1. It indicates SM service to cleanup Session1.

Managing Limiting the Number of Sessions

Enable

By default, this feature is disabled for AM Service and UE Policy Service and enabled for SM service and PCRF Core service on the Policy deployment.

You can enable the Limiting the Number of Sessions functionality using the CNC Console or REST API for Policy.

- **Enable using CNC Console**:
  - To enable the feature for AM Service, set the value of **Enable Max Session Limit** flag to `true` under **Access and Mobility** page.
    For more details on enabling the flag for AM Service, see PCF Access and Mobility.
  - To enable the feature for UE Policy Service, set the value of **Enable Max Session Limit** flag to `true` under **UE Policy Service** page.
    For more details on enabling the flag for UE Policy Service, see PCF UE Policy Service.
  - To enable limiting the number of sessions at user level (for SM service and PCRF Core), set the value of **Session Limit By User** flag to `true` under **Binding Service** page.
    For more details on enabling the **Session Limit By User** flag, see Binding Service.
  - To enable limiting the number of sessions at DNN + SNSSAI level, configure **Max Session Limit By APN** field under **Binding Service** page.
    For more details on enabling and configuring limiting number of sessions at DNN + SNSSAI level using **Max Session Limit By APN** field , see Binding Service.
- **Enable using REST API**:
  - To enable the feature for AM Service, set the value as `true` for **enableLimitingSessionByPerUser** parameter under the System group in the `{apiRoot}/oc-cnpolicy-configuration/v1/services/pcfam` REST API.

- To enable the feature for UE Policy Service, set the value as `true` for **enableLimitingSessionByPerUser** parameter under the System group in the `{apiRoot}/oc-cnpolicy-configuration/v1/services/pcfue` REST API.
- To enable limiting the number of sessions at user level (for SM service and PCRF Core), set the value of **enableSessionLimitByUser** parameter to `true` in `{apiRoot}/oc-cnpolicy-configuration/v1/services/binding` REST API.
- To enable limiting the number of sessions at DNN + SNSSAI level, configure **maxSessionLimitByApn** parameter in `{apiRoot}/oc-cnpolicy-configuration/v1/services/binding` REST API.

For more details on enabling the flag using REST API, see *Converged Policy REST Specifications* section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

You can configure the Limiting the Number of Sessions functionality using the CNC Console or REST API for Policy.

- **Configure using CNC Console**:
  To configure Limiting Number of Sessions feature for AM Service and UE Policy Service, perform the feature configurations under the System group on the respective service configurations page. For more information, see the following sections:
  - PCF Access and Mobility
  - PCF UE Policy Service

  To configure limiting the number of sessions per subscriber and DNN+SNSSAI for SM service and PCRF Core service, configure **Session Limit By User**, **Max Session Limit By User**, and **Max Session Limit By APN** fields under **Binding Service** page. For more information, see Binding Service.
- **Configure using REST API**: Policy provides the following REST API to configure the Limiting the Number of Sessions functionality:
  - **AM service**: {apiRoot}/oc-cnpolicy-configuration/v1/services/pcfam
  - **UE Policy service**: {apiRoot}/oc-cnpolicy-configuration/v1/services/pcfue
  - **Limiting the number of sessions per subscriber and DNN+SNSSAI for SM service and PCRF Core service**: {apiRoot}/oc-cnpolicy-configuration/v1/services/binding

Observe

The following metrics are added to AM Service metrics:

- session_info_request_total
- session_info_response_total

The following metrics are added to UE Policy Service metrics:

- session_info_request_total
- session_info_response_total

For more details on these metrics, see the following sections:

- AM Service Metrics
- UE Service Metrics

Logging

Below is the sample log that can be seen in **pcf-am** once the Max Session limit is hit, and the older session will be deleted.


```
{"instant":
{"epochSecond":1669193988,"nanoOfSecond":609281261},"thread":"pool-4-
thread-15","level":"DEBUG","loggerName":"ocpm.pcf.service.am.core.LimitingSess
ionManager","message":"Limiting Number of Sessions Feature is enabled :
true","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","
threadId":1181,"threadPriority":5,"messageTimestamp":"2022-11-23T08:59:48.609+
0000"}


{"instant":
{"epochSecond":1669193988,"nanoOfSecond":614094447},"thread":"pool-4-
thread-15","level":"DEBUG","loggerName":"ocpm.pcf.service.am.core.LimitingSess
ionManager","message":"Existing Session count :3 exceed max session count
limit:
2","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","thr
eadId":1181,"threadPriority":5,"messageTimestamp":"2022-11-23T08:59:48.614+000
0"}
{"instant":
{"epochSecond":1669193988,"nanoOfSecond":614431105},"thread":"pool-4-
thread-15","level":"DEBUG","loggerName":"ocpm.pcf.service.am.core.LimitingSess
ionManager","message":"The stale association Ids to be deleted are
[5d0ff3a0-0b47-42ab-8fe2-829934465619]","endOfBatch":false,"loggerFqcn":"org.a
pache.logging.slf4j.Log4jLogger","threadId":1181,"threadPriority":5,"messageTi
mestamp":"2022-11-23T08:59:48.614+0000"}


{"instant":
{"epochSecond":1669193991,"nanoOfSecond":624093788},"thread":"pool-4-
thread-16","level":"INFO","loggerName":"ocpm.cne.common.logging.http.HttpLogge
r","marker":{"name":"SUBSCRIBER"},"message":"HttpLog: {type:
'CLIENT_REQUEST', requestId:
'supi;imsi-450081000000002-894f996b-5f94-43f8-9f49-de2753b8c02e', uri:
'http://10.233.74.200:8000/pds/v2/user-data?
reqParam=%7B%22amPolicyDataReq%22%3A%7B%22subscription%22%3Atrue%2C%22resetCon
text%22%3Afalse%2C%22processEtag%22%3Afalse%7D%2C%22ssvEnabled%22%3Afalse%2C%2
2resetContextSSV%22%3Afalse%7D&param=%7B%22user%22%3A%7B%22userIds%22%3A%7B%22
gpsiSet%22%3A%5B%22msisdn-12345678901234%22%5D%2C%22supiSet%22%3A%5B%22imsi-45
0081000000002%22%5D%7D%7D%2C%22request%22%3A%7B%22requestType%22%3A%22DELETE%2
2%2C%22operationType%22%3A%22DELETE%22%2C%22contextOwner%22%3A%22PCF_AM%22%2C%
22contextId%22%3A%225d0ff3a0-0b47-42ab-8fe2-829934465619%22%2C%22notificationI
nfo%22%3Anull%7D%2C%22dnn%22%3A%22NA%22%2C%22gpsi%22%3A%22msisdn-1234567890123
4%22%2C%22sliceinfo%22%3A%7B%22sst%22%3A%22NA%22%2C%22sd%22%3A%22NA%22%7D%2C%2
2supi%22%3A%22imsi-450081000000002%22%7D', method: 'DELETE', headers:
'{subscriber-logging=[supi;imsi-450081000000002, supi;imsi-450081000000002],
Accept=[*/*], User-Agent=[Jetty/9.4.49.v20220914], Host=[10.233.74.200:8000],
Accept-Encoding=[gzip]}', body:
''}","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","t
hreadId":1184,"threadPriority":5,"messageTimestamp":"2022-11-23T08:59:51.624+0
000"}
```

{"instant":
{"epochSecond":1669193991,"nanoOfSecond":647017328},"thread":"pool-4-
thread-17","level":"INFO","loggerName":"ocpm.pcf.service.am.serviceconnector.P
olicyConnector","message":"Policy Request sent to : http://ross-helm-occnp-
pre:8000/v1/policy/engine/pcf-am httpHeader:{subscriber-
logging=[supi;imsi-450081000000002, supi;imsi-450081000000002]} {\"request\":
{\"requestType\":\"AMF\",\"operationType\":\"DELETE\"},\"policyAssociation\":
{\"associationId\":\"5d0ff3a0-0b47-42ab-8fe2-829934465619\",\"createdTimestamp
\":\"2022-11-23T08:59:22.073387212Z\",\"user\":
{\"lastModificationTimestamp\":1669193988612,\"cid\":0,\"userIds\":
{\"LASTACCESSTIME\":\"2022-11-23
08:59:22\",\"PEI\":\"imei-100120010030110\",\"SUPI\":\"imsi-450081000000002\",
\"SITEID\":\"fe7d992b-0541-4c7d-ab84-
c6d70b1b0123\",\"GPSI\":\"msisdn-12345678901234\"},\"amPolicyData\":
{\"lastErrorcode\":null,\"subscCats\":[\"Sliver\"]}},\"policyAssociation\":
{\"request\":{\"notificationUri\":\"http://10.233.14.139:8080/amf/
notify\",\"supi\":\"imsi-450081000000002\",\"gpsi\":\"msisdn-12345678901234\",
\"accessType\":\"3GPP_ACCESS\",\"pei\":\"imei-100120010030110\",\"userLoc\":
{\"nrLocation\":{\"tai\":{\"plmnId\":
{\"mcc\":\"460\",\"mnc\":\"30\"},\"tac\":\"000C27\"},\"ncgi\":{\"plmnId\":
{\"mcc\":\"460\",\"mnc\":\"30\"},\"nrCellId\":\"187900000\"},\"ueLocationTimes
tamp\":\"2019-03-20T14:32:22.222Z\"}},\"servingPlmn\":
{\"mnc\":\"30\",\"mcc\":\"460\"},\"ratType\":\"NR\",\"servAreaRes\":
{\"restrictionType\":\"ALLOWED_AREAS\",\"areas\":[{\"tacs\":
[\"t015\"],\"areaCodes\":\"a014\"}],\"maxNumOfTAs\":1},\"rfsp\":10,\"suppFeat\
":\"0000000000000000\"},\"triggers\":[\"LOC_CH\",\"PRA_CH\"],\"servAreaRes\":
{\"restrictionType\":\"ALLOWED_AREAS\",\"areas\":[{\"tacs\":
[\"12\"],\"areaCodes\":\"123\"}],\"maxNumOfTAs\":1},\"rfsp\":1,\"pras\":
{\"1\":{\"praId\":\"1\",\"trackingAreaList\":[{\"plmnId\":
{\"mcc\":\"111\",\"mnc\":\"11\"},\"tac\":\"1111\"}]}},\"suppFeat\":\"0\"},\"po
licyAssociationRequest\":{\"notificationUri\":\"http://10.233.14.139:8080/amf/
notify\",\"supi\":\"imsi-450081000000002\",\"gpsi\":\"msisdn-12345678901234\",
\"accessType\":\"3GPP_ACCESS\",\"pei\":\"imei-100120010030110\",\"userLoc\":
{\"nrLocation\":{\"tai\":{\"plmnId\":
{\"mcc\":\"460\",\"mnc\":\"30\"},\"tac\":\"000C27\"},\"ncgi\":{\"plmnId\":
{\"mcc\":\"460\",\"mnc\":\"30\"},\"nrCellId\":\"187900000\"},\"ueLocationTimes
tamp\":\"2019-03-20T14:32:22.222Z\"}},\"servingPlmn\":
{\"mnc\":\"30\",\"mcc\":\"460\"},\"ratType\":\"NR\",\"servAreaRes\":
{\"restrictionType\":\"ALLOWED_AREAS\",\"areas\":[{\"tacs\":
[\"t015\"],\"areaCodes\":\"a014\"}],\"maxNumOfTAs\":1},\"rfsp\":10,\"suppFeat\
":\"0000000000000000\"},\"userSubscription\":true,\"sbiBinding\":
{},\"modelObjectId\":\"5d0ff3a0-0b47-42ab-8fe2-829934465619\"},\"user\":
{\"isVisiting\":null,\"userIds\":{\"LASTACCESSTIME\":\"2022-11-23
08:59:22\",\"PEI\":\"imei-100120010030110\",\"SUPI\":\"imsi-450081000000002\",
\"SITEID\":\"fe7d992b-0541-4c7d-ab84-
c6d70b1b0123\",\"GPSI\":\"msisdn-12345678901234\"},\"smPolicyData\":null,\"amP
olicyData\":{\"lastErrorcode\":null,\"subscCats\":
[\"Sliver\"]},\"uePolicySet\":null,\"spendingLimitStatus\":null,\"operatorSpec
ificData\":null,\"shUserData\":null,\"ldapUserData\":null,\"ocsSpendingLimitSt
atus\":null,\"custom\":null}}","endOfBatch":false,"loggerFqcn":"org.apache.log
ging.slf4j.Log4jLogger","threadId":1185,"threadPriority":5,"messageTimestamp":
"2022-11-23T08:59:51.647+0000"}

{"instant":

{"epochSecond":1669193991,"nanoOfSecond":679435548},"thread":"pool-4-thread-18","level":"DEBUG","loggerName":"ocpm.cne.common.db.JdbcTable","message":"trying to execute sql : delete from AmPolicyAssociation where k = ?","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":1190,"threadPriority":5,"messageTimestamp":"2022-11-23T08:59:51.679+0000"}
{"instant":
{"epochSecond":1669193991,"nanoOfSecond":683701171},"thread":"pool-4-thread-18","level":"DEBUG","loggerName":"ocpm.cne.common.db.JdbcTable","message":"deleted 1 records from table AmPolicyAssociation with key 5d0ff3a0-0b47-42ab-8fe2-829934465619","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":1190,"threadPriority":5,"messageTimestamp":"2022-11-23T08:59:51.683+0000"}

Below is the sample log that can be seen in **pcf-ue** once the Max Session limit is hit, and the older session will be deleted.

{"instant":
{"epochSecond":1669195257,"nanoOfSecond":357845539},"thread":"pool-3-thread-9","level":"DEBUG","loggerName":"ocpm.pcf.service.uepolicy.core.LimitingSessionCountManager","message":"The limiting no of sessions feature is enabled","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":1342,"threadPriority":5,"messageTimestamp":"2022-11-23T09:20:57.357+0000"}

{"instant":
{"epochSecond":1669195257,"nanoOfSecond":363242218},"thread":"pool-3-thread-9","level":"DEBUG","loggerName":"ocpm.pcf.service.uepolicy.core.LimitingSessionCountManager","message":"Checking if limit of association Ids is exceeded","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":1342,"threadPriority":5,"messageTimestamp":"2022-11-23T09:20:57.363+0000"}
{"instant":
{"epochSecond":1669195257,"nanoOfSecond":363456742},"thread":"pool-3-thread-9","level":"DEBUG","loggerName":"ocpm.pcf.service.uepolicy.core.LimitingSessionCountManager","message":"The association ids to be deleted are [ocpm.pcf.service.uepolicy.model.UePolicyAssociationImpl@3f172e31]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":1342,"threadPriority":5,"messageTimestamp":"2022-11-23T09:20:57.363+0000"}

{"instant":
{"epochSecond":1669195260,"nanoOfSecond":377243552},"thread":"pool-3-thread-10","level":"INFO","loggerName":"ocpm.pcf.service.uepolicy.serviceconnector.UserServiceConnector","message":"Sent DELETE Request: http://ross-helm-occnp-policy-ds:8000/pds/v2/user-data?
param=%7B%0A%20%20%22user%22%20%3A%20%7B%0A%20%20%20%20%22userIds%22%20%3A%20%7B%0A%20%20%20%20%20%20%22gpsiSet%22%20%3A%20%5B%20%22msisdn-10000000018%22%20%5D%2C%0A%20%20%20%20%20%20%22supiSet%22%20%3A%20%5B%20%22imsi-450081000000018%22%20%5D%0A%20%20%20%20%7D%0A%20%20%7D%2C%0A%20%20%22request%22%20%3A%20%7B%0A%20%20%20%20%22requestType%22%20%3A%20%22DELETE%22%2C%0A%20%20%20%20%22operationType%22%20%3A%20%22DELETE%22%2C%0A%20%20%20%20%22contextOwner%22%20%3A%20%22PCF_UE%22%2C%0A%20%20%20%20%22contextId%22%20%3A%20%22e5c1d501-7761-409a-895c-ed17d224ba99%22%2C%0A%20%20%20%20%22notificationInfo%22%20%3A%20null%0A%20%20%7D%2C%0A%20%20%22dnn%22%20%3A%20%22NA%22%2C%0A%20%20%22sliceinfo%22%20%3A%20%7

B%0A%20%20%20%20%22sd%22%20%3A%20%22NA%22%2C%0A%20%20%20%20%22sst%22%20%3A%20%22NA%22%0A%20%20%7D%2C%0A%20%20%22supi%22%20%3A%20%22imsi-450081000000018%22%2C%0A%20%20%22gpsi%22%20%3A%20%22msisdn-10000000018%22%0A%7D&reqParam=%7B%0A%20%20%22ssvEnabled%22%20%3A%20false%2C%0A%20%20%22resetContextSSV%22%20%3A%20false%0A%7D","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":1349,"threadPriority":5,"messageTimestamp":"2022-11-23T09:21:00.377+0000"}

{"instant":
{"epochSecond":1669195260,"nanoOfSecond":378500887},"thread":"SimpleAsyncTaskExecutor-57289","level":"DEBUG","loggerName":"ocpm.pcf.service.uepolicy.client.LoggingInterceptor","message":"Sending http request:
Request2f16511d\n[\n\tDELETE http://ross-helm-occnp-egress-gateway:8000/namf-comm/v1/ue-contexts/imsi-450081000000018/n1-n2-messages/subscriptions/1234\n\t3gpp-sbi-target-apiroot: http://nf1stub.ross-ns.svc:8080\n\toc-access-token-request-info: {\"targetNfType\":\"AMF\",\"scope\":\"namf-comm\"}
\n\t3gpp-Sbi-Max-Rsp-Time: 3000\n\t3gpp-Sbi-Sender-Timestamp: Wed, 23 Nov 2022 09:21:00.377 GMT\n\tContent-Length: 0\n\tHost: ross-helm-occnp-egress-gateway:8000\n\tConnection: Keep-Alive\n\tAccept-Encoding: gzip\n\tUser-Agent: okhttp/
4.9.3\n]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":68288,"threadPriority":5,"messageTimestamp":"2022-11-23T09:21:00.378+0000"}

{"instant":
{"epochSecond":1669195260,"nanoOfSecond":425336983},"thread":"pool-3-thread-11","level":"INFO","loggerName":"ocpm.pcf.service.uepolicy.serviceconnector.PolicyServiceConnector","message":"Ready to send Policy Request to Policy Engine Service: http://ross-helm-occnp-pre:8000/v1/policy/engine/pcf-ue,
{\"requestType\":\"AMF\",\"operationType\":\"TERMINATE\",\"policyAssociationRequest\":{\"notificationUri\":\"http://10.233.14.139:8080/amcs/v1/uepc-status\",\"supi\":\"imsi-450081000000018\",\"gpsi\":\"msisdn-10000000018\",\"accessType\":\"3GPP_ACCESS\",\"pei\":\"imeisv-9902267900440101\",\"userLoc\":
{\"nrLocation\":{\"tai\":{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"tac\":\"004743\"},\"ncgi\":{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"nrCellId\":\"000000000\"}}},\"servingPlmn\":{\"mnc\":\"05\",\"mcc\":\"450\"},\"ratType\":\"NR\",\"guami\":{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"},\"suppFeat\":\"def\"},\"policyAssociation\":{\"request\":{\"notificationUri\":\"http://
10.233.14.139:8080/amcs/v1/uepc-status\",\"supi\":\"imsi-450081000000018\",\"gpsi\":\"msisdn-10000000018\",\"accessType\":\"3GPP_ACCESS\",\"pei\":\"imeisv-9902267900440101\",\"userLoc\":
{\"nrLocation\":{\"tai\":{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"tac\":\"004743\"},\"ncgi\":{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"nrCellId\":\"000000000\"}}},\"servingPlmn\":{\"mnc\":\"05\",\"mcc\":\"450\"},\"ratType\":\"NR\",\"guami\":{\"plmnId\":
{\"mcc\":\"450\",\"mnc\":\"05\"},\"amfId\":\"010041\"},\"suppFeat\":\"def\"},\"triggers\":[],\"suppFeat\":\"0\"},\"user\":
{\"subscription\":false}}","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","threadId":1353,"threadPriority":5,"messageTimestamp":"2022-11-23T09:21:00.425+0000"}

{"instant":
{"epochSecond":1669195260,"nanoOfSecond":520217022},"thread":"pool-3-thread-12","level":"DEBUG","loggerName":"ocpm.cne.common.db.JdbcTable","messag

```
e":"trying to execute sql : delete from UePolicyAssociation where k
= ?","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","t
hreadId":1356,"threadPriority":5,"messageTimestamp":"2022-11-23T09:21:00.520+0
000"}
{"instant":
{"epochSecond":1669195260,"nanoOfSecond":525313719},"thread":"pool-3-
thread-12","level":"DEBUG","loggerName":"ocpm.cne.common.db.JdbcTable","messag
e":"deleted 1 records from table UePolicyAssociation with key
e5c1d501-7761-409a-895c-
ed17d224ba99","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jL
ogger","threadId":1356,"threadPriority":5,"messageTimestamp":"2022-11-23T09:21
:00.525+0000"}
```


4.95 Handling Stale Data in PDS

Policy supports handling the stale PDS subscriber data for the following services:

- SM Service
- AM Service
- UE Policy Service

When there is no existing session for the subscriber

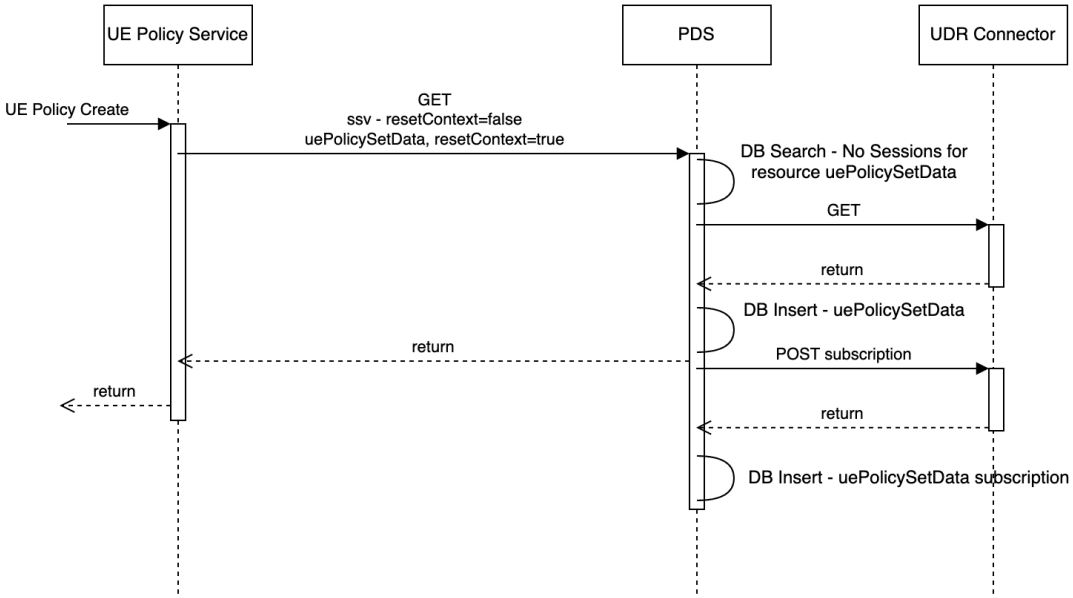
When a policy service receives request for a new session from an external NF, it sends a GET request to PDS to fetch the user or the subscriber data.

If resetContextAmPolicyData or resetContextUePolicySetData flag set to true
If it is a new session for the subscriber, there will be no session information for the subscriber in the respective policy service database. While sending the GET request to PDS, the policy service sets the value of the `resetContextAmPolicyData` or `resetContextUePolicySetData` flag to `true`.

PDS sends a GET request to UDR Connector to fetch the user or the subscription data for the subscriber. According to the response, PDS updates the subscriber data in PolicyDS database.

Also, as there is no existing session information in PolicyDS database, PDS sends a POST subscription request to UDR Connector to create a subscription. After it receives a response from the UDR Connector, PDS inserts the subscription details to its database and returns the same to the policy service.

Figure 4-159 resetContextUePolicySetData set to true when there are no existing sessions for UE Policy service



When there are existing sessions for the subscriber

When a policy service receives request for a new session from an external NF, it sends a GET request to PDS to fetch the SSV and the subscription data.

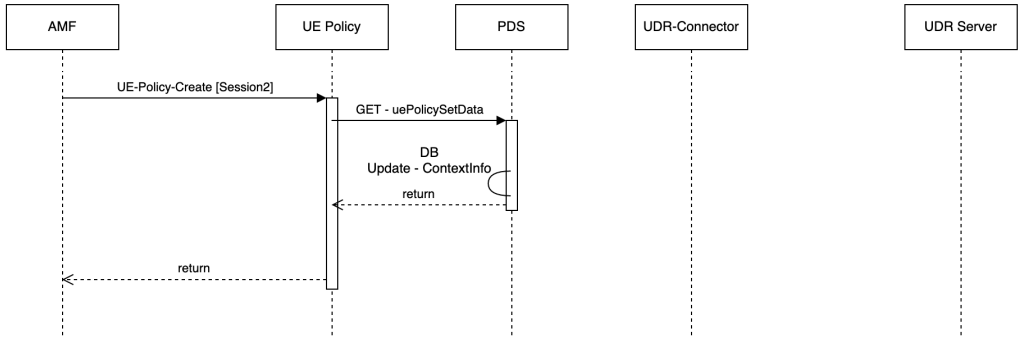
If resetContextAmPolicyData or resetContextUePolicySetData flag set to false
If there are existing sessions for the subscriber, while sending the GET request to PDS, the policy service sets the value of the `resetContextAmPolicyData` or `resetContextUePolicySetData` flag to `false`.

As `resetContextAmPolicyData` or `resetContextUePolicySetData` flag is set to `false`, PDS does not attempt to get the data from UDR Connector.

PDS updates the PolicyDS database as per the request received from the policy service. It responds to the policy service with the data present in PolicyDS database.

If `CONCURRENT_REQUEST_GUARD_TIME` is configured, PDS removes the sessions which are older than the guard time.

Figure 4-160 resetContextUePolicySetData set to false when there are existing sessions for UE Policy service



Revalidation of the subscriber data in PDS

If the revalidation configuration is enabled for PDS, that is, if `Enable Fetch and Resubscribe` is set to `true`, and there is already existing information for the subscriber in its database, if the number of sessions exceeds the `Max Sessions Count`, PDS sends a GET request to re-fetch the subscription data from UDR Connector.

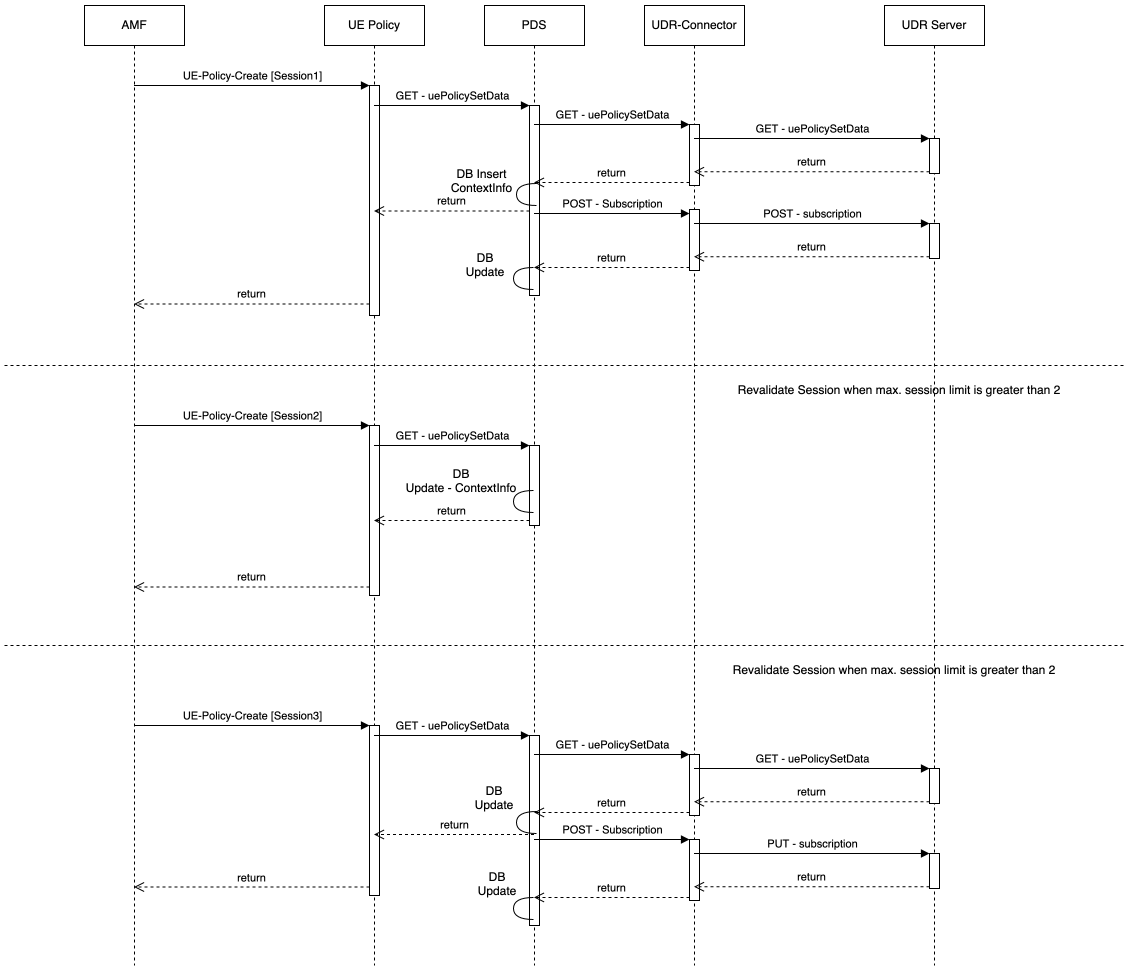
PDS sends a PUT subscription request to UDR Connector with the stored subscription details from its PolicyDS database. If the subscription details are validated successfully, UDR Connector responds with a 200 ok message.

If the validation fails, UDR Connector responds with a 404 response. PDS sends a POST subscription request to UDR Connector to create a subscription. UDR Connector forwards the POST subscription request to UDR Server. After UDR Server responds to UDR Connector, UDR Connector responds to PDS with the subscription details received from UDR Server. PDS inserts the subscription details to the PolicyDS database and returns the same to the policy service.

Example:

Here is an example call flow for revalidation of the subscription data for UE Policy Service sessions.

Figure 4-161 Revalidation of the subscription data for UE Policy Sessions



Managing Handling PDS Stale Sessions

Enable

You can enable `resetContextAmPolicyData` and `resetContextUePolicySetData` flags in `custom-values.yaml` file.

For more information about the resetContext flags for AM Service and UE Policy Service, see *Customizing Policy* chapter *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*.

To enable the PDS revalidation functionality:

You can enable the PDS revalidation functionality using the CNC Console or REST API for Policy.

- **Enable using CNC Console**:
  To enable the PDS revalidation functionality using the CNC Console, set the value of **Enable Fetch and Resubscribe** flag to `true` in the **PDS** page.

  For more details about **Enable Fetch and Resubscribe** flag, see PDS.

- **Enable using REST API**:
  To enable the PDS revalidation functionality, set the value as `true` for **enableFetchAndResubscribe** parameter under the Common group in the `{apiRoot}/oc-cnpolicy-configuration/v1/services/pds/pdsSettings` REST API.

  For more details on enabling the flag using REST API, see *Policy REST Specifications* section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

Configure

You can configure `resetContextAmPolicyData` and `resetContextUePolicySetData` flags in `custom-values.yaml` file.

For more information about the resetContext flags for AM Service and UE Policy Service, see *Customizing Policy* chapter *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*.

To enable the PDS revalidation functionality:

You can configure the PDS revalidation functionality to handle the stale PDS data using the CNC Console or REST API for Policy.

- **Configure using CNC Console**:
  To configure PDS revalidation functionality in PDS, perform the feature configurations on **PDS** page. For more details, see PDS.
- **Configure using REST API**: Policy provides {apiRoot}/oc-cnpolicy-configuration/v1/services/pds/pdsSettings REST API to configure the PDS revalidation functionality.

> **Note**
>
> In case of PDS subscriber data revalidation for UE and AM policy services, the value of DNN, SD and SST parameters must be `NA`.

Observe

The following metrics are used to support the PDS revalidation functionality:

- remove_contextInfo_total
- revalidation_request
- revalidation_response

For details on the metrics, see Policy DS Metrics.

Logging

Below is the sample log from Policy-ds when MaxSessionLimit is hit for revalidation.

```
{"instant":
{"epochSecond":1669192175,"nanoOfSecond":908706032},"thread":"boundedElastic-2
9","level":"DEBUG","loggerName":"ocpm.uds.policyds.workflow.flowtask.dbtask.Ab
stractDBFlowTask","message":"Comparing Existing Dnn Slice count Map :
{NA:NA:NA=1} with Configured Max Dnn Slice Count
        Map : {NA:NA:NA=1}
","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","thre
adId":7491,"threadPriority":5,"messageTimestamp":"2022-11-23T08:29:35.908+0000
```

```
"}{"instant":
{"epochSecond":1669192175,"nanoOfSecond":909143676},"thread":"boundedElastic-2
9","level":"INFO","loggerName":"ocpm.uds.policyds.workflow.flowtask.dbtask.Abs
tractDBFlowTask","message":"Revalidation is required so going to fetch
updated data for source type:

SpendingLimitData","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.L
og4jLogger","threadId":7491,"threadPriority":5,"messageTimestamp":"2022-11-23T
08:29:35.909+0000"}  {"instant":
{"epochSecond":1669192175,"nanoOfSecond":910933620},"thread":"boundedElastic-2
9","level":"DEBUG","loggerName":"ocpm.uds.policyds.workflow.flowtask.dbtask.Ab
stractDBFlowTask","message":"Comparing Existing Dnn Slice count Map :
{NA:NA:NA=1} with Configured Max Dnn Slice Count
        Map : {NA:NA:NA=1}
","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","thre
adId":7491,"threadPriority":5,"messageTimestamp":"2022-11-23T08:29:35.910+0000
"}{"instant":
{"epochSecond":1669192175,"nanoOfSecond":911364865},"thread":"boundedElastic-2
9","level":"INFO","loggerName":"ocpm.uds.policyds.workflow.flowtask.dbtask.Abs
tractDBFlowTask","message":"Revalidation is required so going to fetch
updated data for source type:

AmPolicyData","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jL
ogger","threadId":7491,"threadPriority":5,"messageTimestamp":"2022-11-23T08:29
:35.911+0000"} {"instant":
{"epochSecond":1669192175,"nanoOfSecond":915496199},"thread":"boundedElastic-2
9","level":"INFO","loggerName":"ocpm.uds.policyds.rest.client.WebClientService
Impl","message":"Req
        body{\"amPolicyDataReq\":
{\"subscription\":false,\"params\":null,\"resetContext\":false,\"processEtag\"
:false},\"ssvEnabled\":false,\"resetContextSSV\":false}
        sent to URL http://ross-helm-occnp-udr-connector:8000/userservice/user-data/
imsi-450081000000024,msisdn-12345678901234 with
        header {uber-trace-id=[a398635f494761ab:a398635f494761ab:0:0,
        37b688bf4531ac85:37b688bf4531ac85:0:0],
          oc-data-source-route-info=[{\"amPolicyData\":{\"dataSourceInfo\":
{\"dataSourceId\":\"fe7d992b-0541-4c7d-ab84-555552222222:94d8d19a-
b0d9-4d6d-994f-39a49ed5c111\",\"host\":\"nf1stub.ross-
ns.svc\",\"port\":8080},\"subscriptionResourceInfo\":
{\"subscriptionId\":\"http://nf1stub.ross-ns.svc:8080/nudr-dr/v1/policy-data/subs-to-notify/
1\",\"host\":\"nf1stub.ross-ns.svc\",\"port\":8080,\"subscriptionData\":\"{\\
\"policyDataSubscription\\\":{\\\"notificationUri\\\":\\\"https://ross-helm-occnp-
ingress-gateway:443/udrservice/notification/imsi-450081000000024\\\",\\
\"monitoredResourceUris\\\":[\\\"http://nf1stub.ross-ns.svc:8080/nudr-dr/v1/policy-data/ues/
imsi-450081000000024/am-data\\\"],\\\"supportedFeatures\\\":\\\"f\\\"}}
\"},\"nfBindingInfo\":{},\"otherAttr\":{\"processETag\":false}}}]}

","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","thre
adId":7491,"threadPriority":5,"messageTimestamp":"2022-11-23T08:29:35.915+0000
"} {"instant":
{"epochSecond":1669192175,"nanoOfSecond":922192005},"thread":"boundedElastic-2
9","level":"INFO","loggerName":"ocpm.uds.policyds.rest.client.WebClientService
Impl","message":"Req bodynull sent to URL http://ross-helm-occnp-chf-connector:8000/
chfservice/spending-limit/imsi-450081000000024?
gpsi=msisdn-12345678901234&plmn.mcc=460&plmn.mnc=30&asyncQuery=false&resetContext=false
with
```

```
        header {uber-trace-id=[a398635f494761ab:a398635f494761ab:0:0,
        37b688bf4531ac85:37b688bf4531ac85:0:0,
bc38f03768302c41:bc38f03768302c41:0:0],
        3gpp-Sbi-Max-Rsp-Time=[2700],
          oc-data-source-route-info=[{\"spendingLimitStatus\":
{\"dataSourceInfo\":{\"dataSourceId\":\"fe7d992b-0541-4c7d-
ab84-666666666666:chf1\",\"host\":\"nf1stub.ross-
ns.svc\",\"port\":8080},\"subscriptionResourceInfo\":
{\"subscriptionId\":\"http://nf1stub.ross-ns.svc/nchf-spendinglimitcontrol/v1/subscriptions/
1\",\"host\":\"nf1stub.ross-ns.svc\",\"port\":8080},\"nfBindingInfo\":
{},\"otherAttr\":{}}}]}

","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","thre
adId":7491,"threadPriority":5,"messageTimestamp":"2022-11-23T08:29:35.922+0000
"}
```


4.96 Support for User-Agent Header

User-Agent header helps the producer Network Function (NF) to identify the consumer NF that has sent the request. To implement this, 3GPP introduced the use of User-Agent header for consumers to include the same in service requests. Additionally, producers may require to support the validation of the User-Agent headers to complete the request identification process in the network.

With the integration of this feature, User-Agent header helps the producer Network Function (NF) to identify the consumer NF that has sent the request.

The following format is used to generate User-Agent header:

<NF Type>-<Instance-Id> <FQDN>

where, `<NF Type>` is the type of the Network Function.

`<Instance-Id` is the instance ID of the NF.

`<FQDN>` is the FQDN of the NF.

Example: `PCF-54804518-4191-46b3-955c-ac631f953ed8 pcf1.east.5gc.mnc012.mcc234.3gppnetwork.org`

There is no configuration required at PCF for incoming user-agent header. By default, PCF validates the format of the user-agent header and updates the ingress metrics.

When the User-Agent header is missing in the incoming requests sent to PCF then the corresponding metric will not be able to retrieve information from whom it is requesting the service. Nonetheless, the request will be fulfilled without any issues.

For the generation of the User-Agent header when it is enabled through Helm configurations and header information is missing, then it will be picked from the OAuthClient module to generate the header.

If the User-Agent header is present in the request towards Policy, then the value present in the header is overwritten or forwarded based on the `overwriteHeader` flag. If the flag is set to true, then the header is overwritten.

> **Note**
>
> `nfInstanceId` and `nfType` are picked from `userAgentHeader` configuration, if available. Otherwise, they are picked from `oauthClient` configuration. This does not require oauthClient to be enabled but mandates oauth client configuration to be present. If `nfInstanceId` and `nfType` are not available in both oauth client configurations and in User-Agent configuration, then the startup probe fails.

The header is applied to every outgoing message generated from Egress Gateway that includes the SMF, UE, and AM interfaces.

Managing Support for User-Agent Header

Enable

You can enable the User-Agent Header feature using REST or Helm configuration.

- Helm: Set the value of the parameter `userAgentHeaderValidationConfigMode` to Helm in the custom-values.yaml file. For more information, see the *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*.
- REST API: Set the value of the parameter `userAgentHeaderValidationConfigMode` to REST in the custom-values.yaml file. REST configuration from the JSON bodies sent to path: "*/pcf/nf-common-component/v1/egw/useragentheader*" is stored in a database under the common_config table. For more information, see the *Oracle Communications Cloud Native Core, Converged Policy REST Specifications Guide*.

Configure

You can configure the User-Agent Header using REST or Helm.

Following is a sample configuration for User-Agent Header in custom-values.yaml.file:

```
userAgentHeaderConfigMode: HELM
userAgentHeader:
  enabled: false # flag to enable or disable the feature
  nfType: "PCF" # NF type of consumer NF
  nfInstanceId: "2d8e8e68-24ad-11ed-861d-0242ac120002" # NF type of consumer
NF
  addFqdnToHeader: true # Flag to add fqdn. If enabled then user-agent header
will be
generated along with the fqdn configured otherwise fqdn will not be added
  nfFqdn: "oracle1.pcf.pacific.org" #fqdn of NF. This is not the fqdn of
gateway
  overwriteHeader: true
```

To configure the User-Agent header at Egress Gateway using REST API, see user-Agent Header in *Oracle Communications Cloud Native Core, Converged Policy REST Specifications Guide*.

Observe

Policy provides the following metric specific to User-Agent Header feature:

- oc.ingressgateway.http.requests

For more information, see User-Agent Header Metrics section.

4.97 Support for Spending Limit Status Reporting

Policy provides an option to enable spending limit status in each of the create, update, and delete requests from the Policy Services (SM Service, AM Service, and UE Policy Service).

The Spending Limit control service enables the Policy services to retrieve policy counter status information per subscriber from CHF.

That is, by subscribing to spending limit reporting (CHF Data), Policy services can receive notifications on Policy Counters from CHF.

PolicyDS handles the cases when multiple services subscribe to spending limit reporting as follows:

When one service subscribes to a given policy counter, another service subscribes for another policy counter, then PDS will take the policy counters from the last request and both services will be subscribed to receive notifications for those policy counters.

For example:

If UE Policy Service subscribes for PC1 and PC2 and AM Service subscribes for PC3, both UE Policy Service and AM Service will receive notification only for PC3.

Whenever there is create, update, and delete requests from the Policy Services (SM Service, AM Service, and UE Policy Service), Policy checks if CHF Data flag is enabled. If the CHF Data flag is enabled, it allows interaction between PDS and CHF through CHF Connector to retrieve the CHF data,

Policy allows subscription to Spending Limit Control Service (Nchf_SpendingLimitControl_Subscribe), which provides notification of changes in the status of the policy counters available and to retrieve the status of the policy counters for which the subscription is accepted.

That is, it provides:

- initial spending limit retrieval
- intermediate spending limit report retrieval

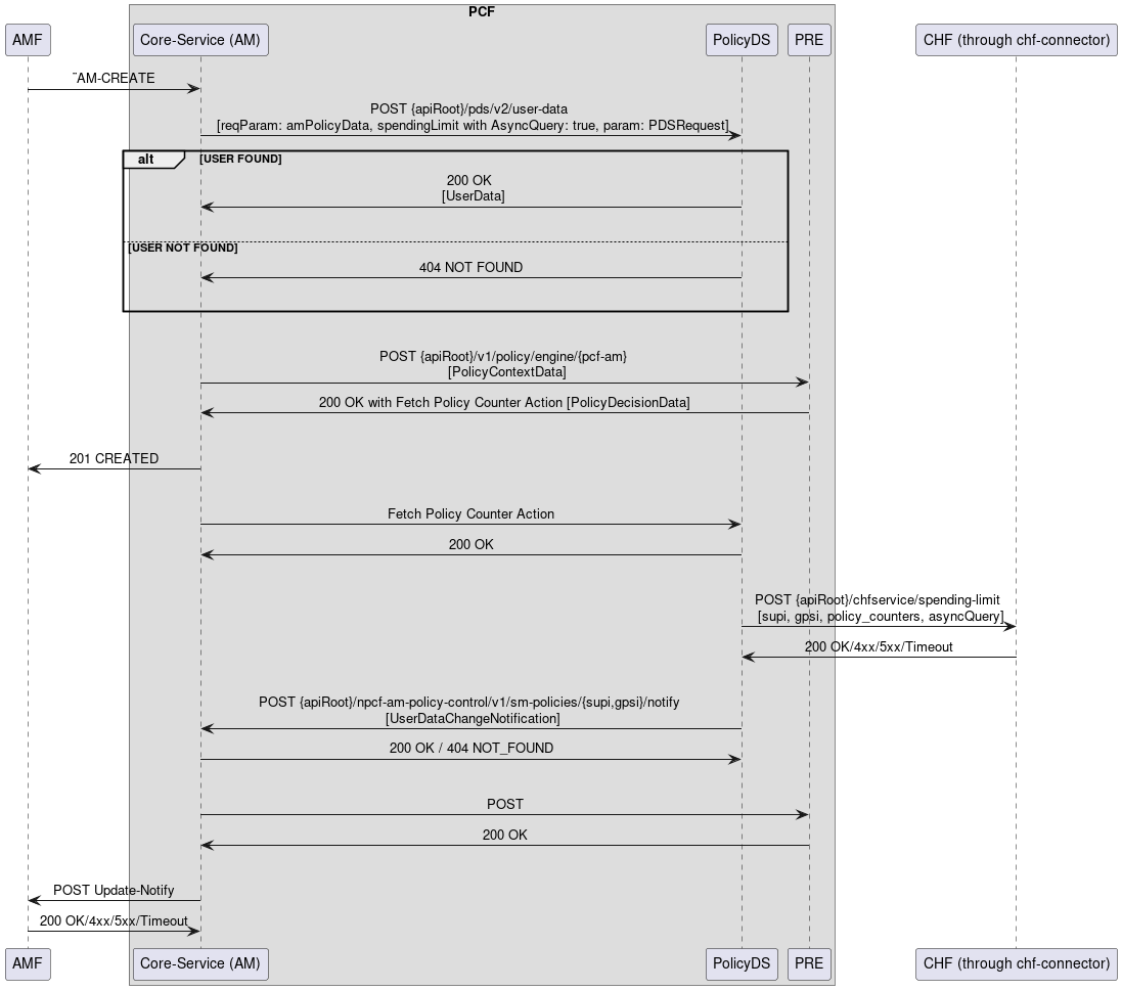
Also, Policy allows asynchronous interaction between PDS and CHF when Async CHF Query Enabled flag is set to true.

> **Note**
>
> This feature supports only Asynchronous flow.

Call Flow

Below is an example for retrieving CHF Data for AM Service:

Figure 4-162 Spending limit status report for AM Service



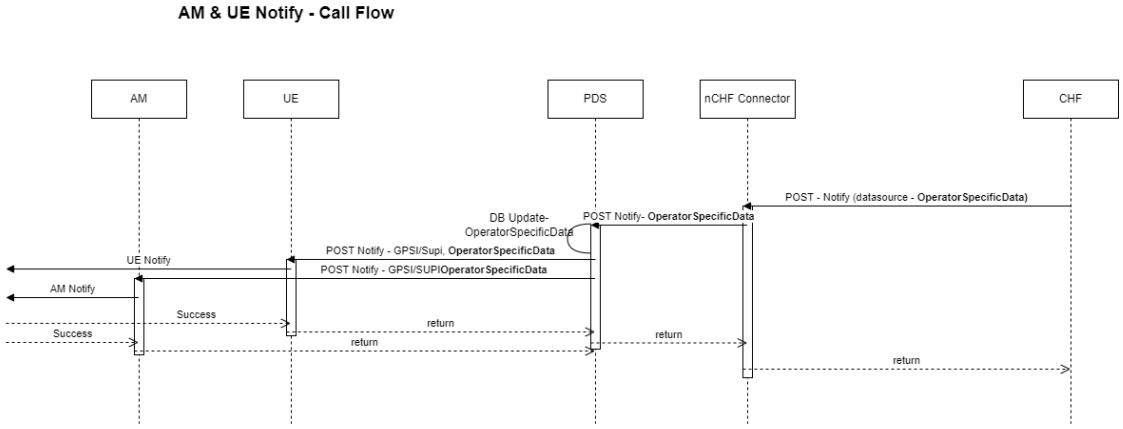
Subscription create:

1. AM Service receives request for a new session from AMF.
2. AM Service checks if CHF Data flag is enabled.
3. AM Service sends a GET request to PDS to fetch the user data. If Async CHF Query Enabled flag is also enabled, it sets AsyncQuery to true in the request parameters.
4. If the user data is present in PolicyDS database, PDS responds to AM Service with 200 OK message.
5. If the requested data is not present in PolicyDS, PDS responds to AM Service with a 404 NOT FOUND message. PDS sends a request to UDR through UDR Connector to fetch the UserData.
6. AM Service, sends a POST request to PRE. Using the fetch policy counter blockly AM identifies the policy counters for which the information is requested.
7. After receiving a 200 OK response from PRE, AM Service sends a 201 created message to AMF.
8. AM sends another request to PDS of type REQUEST-PC. Since this REQUEST-PC flow is asynchronous, PDS responds immediately with a 204 OK message to AM.

9. In asynchronous way, PDS contacts CHF through CHF Connector to subscribe for the policy counters that PRE returned to AM Service.
10. After receiving the information from CHF, PDS sends a notification to AM Service with the SpendingLimit data.
11. AM Service processes this information and if needed it sends an update notify to AMF Service.

Notification call flow from CHF to AM and UE

Figure 4-163 Update notification to AM Service and UE Policy Service



Whenever CHF detects that the status of a policy counter(s) has changed, it sends a notification to PDS.

1. The change notification can be generated for any of the following reasons.
   a. Change in status of Policy counters.
   b. Future change in status of policy counters along with time at which it will change.
   c. Policy counter identifier is no longer applicable to subscriber.
2. The update notification can include:
   - SUPI
   - GPSI
   - Policy counter status as event information
   - Pending policy counter statuses and their activation times as event information
3. PDS updates its database and sends a POST request to AM Service or UE Service.
4. AM Service or UE Policy Service processes the notification and if required it sends an update notify to AMF.
5. AM Service and UE Policy Services accepts the message and responds with a 204 No content message back to CHF through PDS and CHF connector.

Intermediate spending limit report transaction from the CHF

CHF provides an intermediate spending limit report transaction containing pending policy counter statuses and their activation times for a previously provided policy counter.

1. PDS forwards the details to AM Service and UE Policy Service. The Policy services replaces the existing pending policy counter statuses and their activation times if any in their respective databases.
2. If the intermediate spending limit report transaction from the CHF includes no pending policy counter statuses and their activation times are included for a policy counter, the Policy services cancels all the previously provided pending policy counter statuses and their activation times for that particular policy counter.

Managing Handling PDS Stale Sessions

Enable

By default, Spending Limit Status Report feature is not enabled on the Policy deployment.You can enable this functionality using the CNC Console.

To enable this feature for AM Service, enable **CHF Data** flag under **User group** on the **PCF Access and Mobility** page.

To enable this feature for UE Policy Service, enable **CHF Data** flag under **User group** on the **PCF UE Policy** page.

For more details on enabling the flag for AM Service, see PCF Access and Mobility.

For more details on enabling the flag for UE Policy Service, see PCF UE Policy Service.

Configure

You can configure the Spending Limit Status Report feature using the CNC Console.

To configure the feature for AM and UE services, perform the feature configurations under the **User** group on the respective service configurations page. For more information, see the following sections:

- PCF Access and Mobility
- PCF UE Policy Service

Observe

The following metrics are added to PDS Service metrics:

- server_request_total
- server_response_total

For more details on these metrics, see Policy DS Metrics.

4.98 NF Scoring for a Site

The NF Scoring feature calculates the score for a site based on Network Function (NF) specific factors such as metrics, alerts, etc. The NF Scoring feature helps the operator to determine the health of a site as compared to other sites. Comparing the NF scores within or across the sites helps the customers to choose the site.

One of the use cases is the Controlled Shutdown feature that allows the operator to partially or completely isolate the site. The NF Scoring feature helps the operators to choose which site to partially or completely isolate based on NF scoring.

App-Info service queries and calculates NF-Score as it has the site information.

App Info Scoring Mechanism:

App Info reads the configurations from the common config server to check if NF Scoring functionality is enabled or not. It works in the following ways:

- **Continuous NF Score Calculation:** When the NF Scoring feature is enabled, app info periodically reads the configurations to calculate the score.
- **On-Demand NF Score Calculation:** When the NF Scoring feature is enabled, app info fetches all the factors or criteria to calculate the NF Score. It is real-time fetching of factors and then the NF score is calculated on demand.

Table 4-66 NF Scoring Criteria

Factors	Default Score	Formula to calculate Factor Score	Details
TPS	20	min(Current-TPS / Max-TPS * Max- TPS-Score, Max- TPS-Score)	Current-TPS = IGW + EGW + Diameter Ingress + Diameter Egress Max-TPS specifies the maximum TPS. Max- TPS-Score Specifies the maximum score of the TPS.
Service	30	A / N * Max-SVC-Score	A = Number of available services N = Number of configured services Max-SVC-Score Specifies the maximum score of the Service Health.
Connection	20	min(Conn-Current / Conn-Total * Conn-Score, Conn-Score)	Conn-current specifies the number of connections from network to Policy. Conn-Total specifies the total number of connections expected from network to Policy. Conn-Score specifies the score for the connection.
Replication-health	30	min(Site-Current / Site-Total * Site-Score, Site-Score)	Site-Total specifies the total number of possible replication links. Site-Current specifies the available active healthy links. Replication-health Score specifies the score for the replication-health.
Locality-Preference	5	NA	The value of Locality-Preference is added for NF score calculation.
Critical-Alerts	2	CrN * Configured-Score	CrN is the Number of active critical alarms. Configured-Score-Critical-Alerts specifies the score configured by the user.

Table 4-66 (Cont.) NF Scoring Criteria

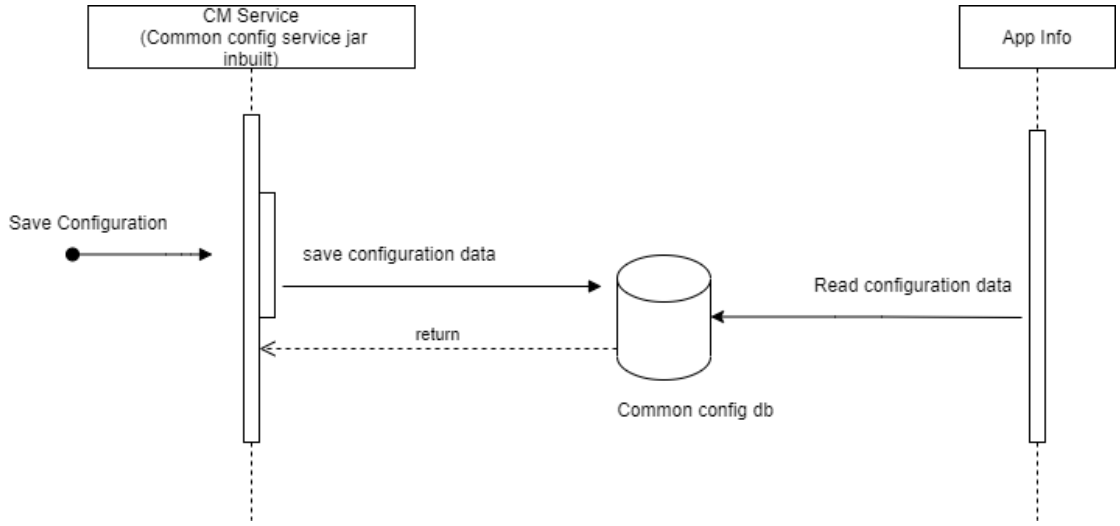
Factors	Default Score	Formula to calculate Factor Score	Details
Major-Alerts	1	MaN * Configured-Score	MaN is the Number of active Major alarms. Configured-Score-Major-Alerts specifies the score configured by the user.
Minor-Alerts	0	MiN * Configured-Score	MiN is the Number of active Minor alarms. Configured-Score-Minor-Alerts specifies the score configured by the user.

Formula for NF scoring of a site: Sum of TPS-Score, SVC-Score, Conn-Score, Replication-health, and Locality-Preference score subtracted by Alerts scores.

Call Flows

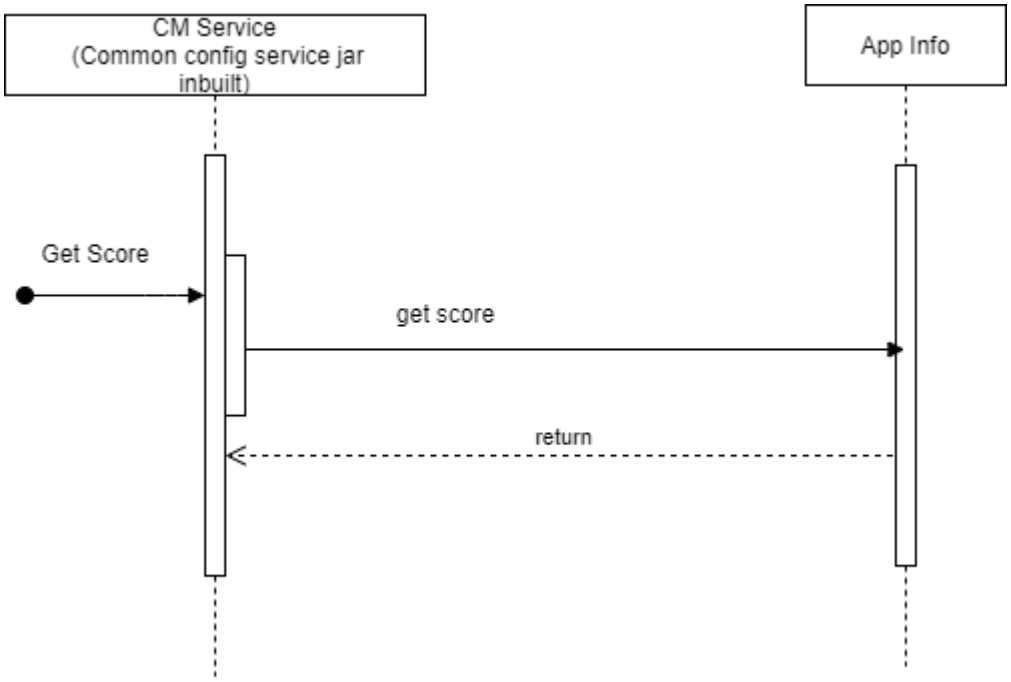
This section describes examples of the call flows for the NF Scoring feature:

Figure 4-164 Call flow to Save Configuration Data



The operator sends a request to save the configuration is sent to the CM service. It saves the configuration data to the common config database. App-Info reads the configuration data and returns the acknowledgment.

Figure 4-165 Call flow to Get the NF Score



The operator sends a request to CM service to get the score. CM service requests it to App-Info. App-Info queries and calculates NF-Score.

Managing NF Scoring for a Site

Enable

You can enable this feature by selecting the **Enable NF Scoring** field in the **Settings** page of **NF Scoring**.

For more information about enabling the feature through CNC Console, see NF Scoring Configurations .

Configure

Configuring using CNC Console: The NF Scoring feature can be configured through CNC Console. For more information, see NF Scoring Configurations.

> **Note**
>
> You can configure the env variable, NF_SCORING_INTERVAL, in deployment of app-info. Default value is 30 seconds (changing the env variable would result into restart of app-info pod).

Configuring using REST API: Policy provides the following REST API for configuration: `/oc-cnpolicy-configuration/v1/ nfscore.`

You can perform the GET operation to get NF score. For more information about REST API configuration, see "NF Scoring for a Site" in Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide.

Observe

Metrics

Policy provides the following metrics specific to NF Scoring feature:

- nfscore
- nfScoringFactorActualValue

For more information, see AppInfo Metrics.

4.99 Consistent UDR Updates Using ETag

When two different Policies are connected to UDR in a georedundant setup and both are updating the UsageMonData for the same subscriber simultaneously, there is the possibility of data loss as the update can overwrite each other. ETag (Entity Tag) support helps to make sure that the update is successful only when the consumer has the latest set of data.

For more information on how to configure this feature using CNC Console, see the Configuring Usage Monitoring.

4.100 Support for Resource Allocation for PCC Rules

When Successful Resource Allocation (SRA) is enabled, Policy provides the status of the Policy and Charging Control Rules (PCC Rules) that are successfully installed and validated by Session Management Function (SMF).

Policy writer can make the appropriate action based on the successful activation of a rule.

For example, a Policy writer can hold updating the installed PCC rule unless SRA is applied at SMF.

> **Note**
>
> This feature is applicable to the PCC Rules that are installed by Policy and not through Applicaiton Function (AF).

To apply the SRA functionality, PCC Rule must include `policyCtrlReqTriggers` and `lastReqRuleData`.

For example:

```
"policyCtrlReqTriggers":[
   "PLMN_CH",
   "UE_IP_CH",
   "DEF_QOS_CH",
   "AC_TY_CH",
   "SUCC_RES_ALLO"
],
"lastReqRuleData":[
   {
      "refPccRuleIds":[
         "0_2",
```

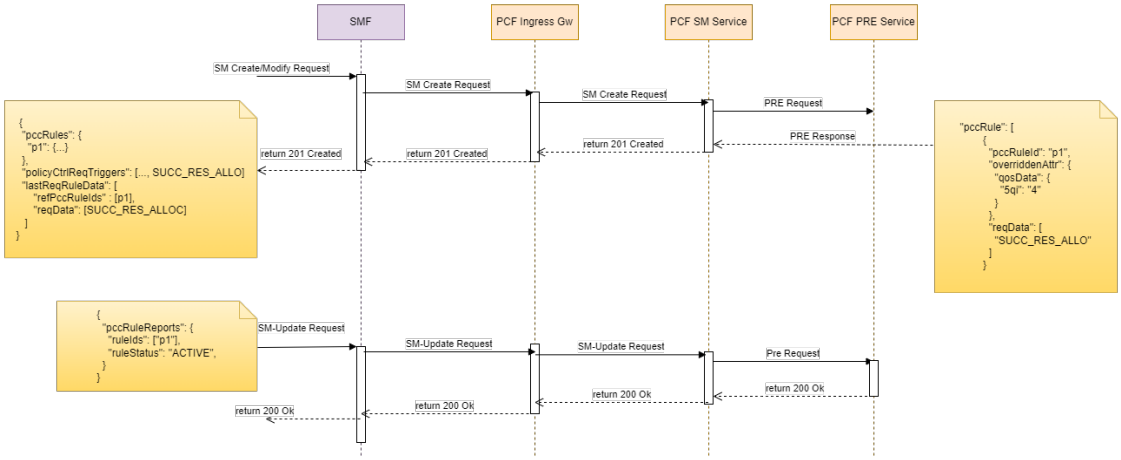
```
            "0_3"
         ],
         "reqData":[
            "SUCC_RES_ALLO"
         ]
      }
]
```

In the above example, `SUCC_RES_ALLO` under `policyCtrlReqTriggers` triggers the application of SRA. `lastReqRuleData` includes the last requested rule data and `SUCC_RES_ALLO` under `reqData` is the requested data.

> **Note**
>
> Policy does not have to send any `policyCtrlReqTriggers` to receive failed-to-install PCC rules from SMF.

Call Flow for SRA

Figure 4-166 PCC Rule for SM Create and SM Update requests

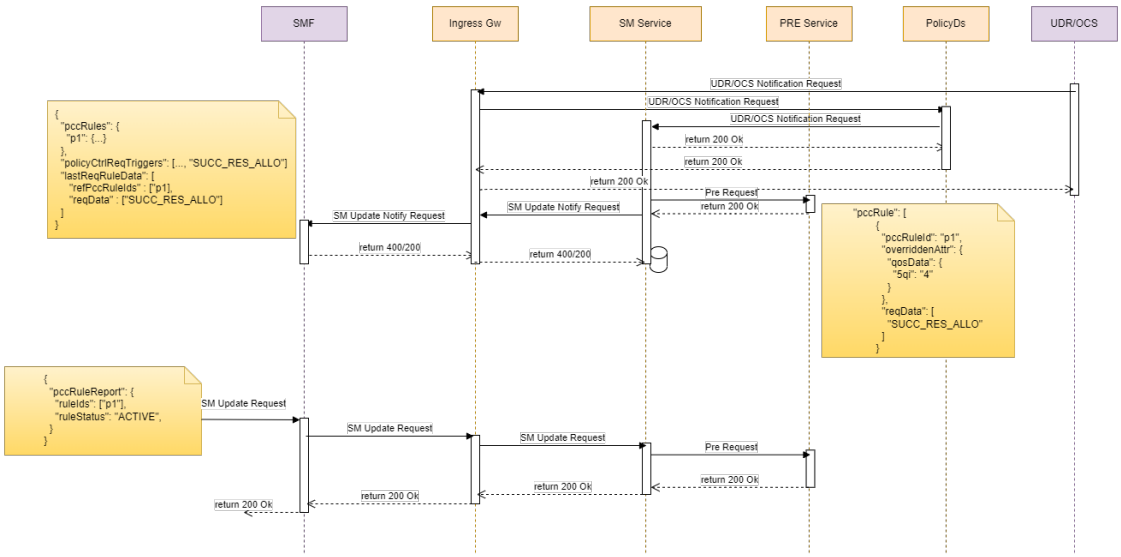


1. SMF receives an SM Create request with the PCC Rule installation information to create a new PCC Rule p1 with requested data as `SUCC_RES_ALLO` indicating SRA.
2. SMF sends the SM Create request to SM Service via Ingress Gateway.
3. SM Service forwards the SM Create request to PRE for evaluation.
4. PRE evaluates the details and responds to SM Service.
5. SM Service creates the session details in the session database, sends a 201 created message to SMF via the Ingress Gateway.

> **Note**
>
> The SM Create contains different media components. If any media component that contains a flow status set as 'removed', the notification towards SMF will not be sent.

6. SMF installs and validates PCC Rule p1. It includes `SUCC_RES_ALLO` under PolicyctrlReqTrigger indicating the application of SRA, along with the details of the last requested rule data and requested data containing `SUCC_RES_ALLO`.
7. After the session creation, SMF receives an SM Update request that includes PCC rule report for p1 with `ruleStatus` as `ACTIVE`.
8. SMF sends the SM Update request to SM Service via Ingress Gateway.
9. SM Service forwards the SM Create request to PRE for evaluation.
10. SM Service forwards the 200 ok response that it receives from PRE to SMF via the Ingress Gateway.
11. SMF validates the details and responds with a 200 ok message.

Figure 4-167 PCC Rule in SM-UPDATE-NOTIFY request (Re-authorization)



1. Policy receives an SM Update-Notify from one of the data sources such as UDR via the ingress gateway.
2. Policy sends the update notification request to SM Service.
3. SM Service acknowledges the update by returning a 200 ok message to Policy.
4. Policy in-turn responds to the data source with a 200 ok message via the ingress gateway.
5. SM Service shares the SM Update Notify details to PRE for evaluation. The update-notify information includes an update to the PCC Rule p1 with `overriddenAttr` containing the updates to the PCC Rule.
6. PRE evaluates the update and returns a 200 ok message to SM Service.
7. SM Service sends the SM Update Notify request to SMF via the Ingress Gateway.

8. SMF modifies and validates the PCC Rule p1 as per the given details and returns a 400/200 message to SM Service via the Ingress Gateway.
9. SM Service updates the session database accordingly.
10. After the successful processing of the SM Update-Notify request, SMF receives a SM Update request containing PCC rule report for P1 with `ruleStatus` as `ACTIVE`.
11. SMF forwards the SM Update request to SM Service via the Ingress Gateway.
12. SM Service shares the SM Update with PRE for evaluation and receives a 200 ok response.
13. SM Service sends the 200 ok response to SMF via the Ingress Gateway.

Managing Support for Successful Resource Allocation for PCC Rules

Enable

By default, the SRA support for PCC Rules is disabled. You can enable this functionality using the CNC Console.

To enable this feature, under **Rule** section in **PCF Session Management** page in CNC Console, set the value of `Default PCC Rule Requested Rule Data` parameter to `SUCC_RES_ALLO`.

For more details on enabling the SRA support for PCC Rules, see Settings.

Configure

You can use blockly action to configure Install/Update/Remove PCC Rules with Override Attributes. For more information, see *PCF-SM Category* in *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.

Observe

Added `rule_action` metric to support Successful Resource Allocation.

For more information on metrics, see SM Service Metrics.

4.101 Subscriber State Variables in Policy

State variables hold the intermediate state of the policy evaluation. The state variables are defined by the policy writer that can be set within a policy action to be used at a later time during policy rule execution (in either conditions or actions). The names of these variables are not predefined and are determined by the policy writer.

State variables have `scope` which determines how long the value is persisted after it is set.

Depending on the scope, the following state variables are defined in Policy:

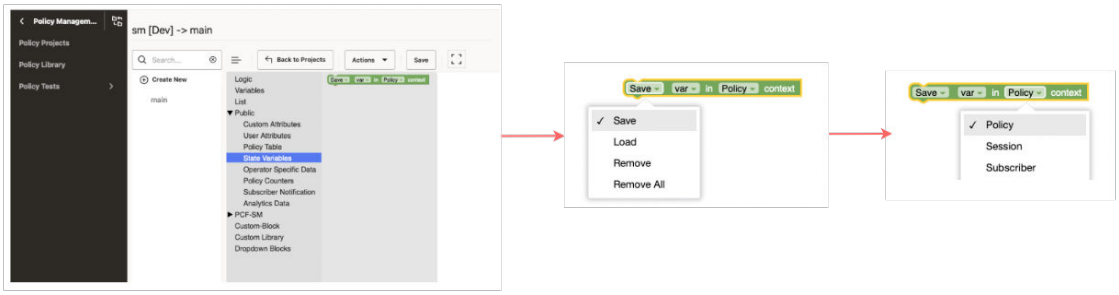
- **Policy Evaluation State Variables**: Used during Policy evaluation. These variables exist only for the lifetime of a policy evaluation cycle (the process of evaluating all the policies for a single request/context).
- **Session State Variables**: Contains a value that is saved as long as the session they are associated with is valid. These variables are stored in the session database in the respective core services (SM Service, AM Service, UE Policy Service, and PCRF Core). On the subsequent operation of the session, the values are extracted and reused. Once the session is terminated, these variables will no longer have a value and will no longer be available for use in policies.

- **Subscriber Local Policy Evaluation State Variables**: Created for a subscriber and remains until at least one session for the subscriber exists. When there are multiple sessions associated with a subscriber across services (SM Service, AM Service, UE Policy Service, or PCRF Core), these subscriber state variables are used across sessions based on the subscriber ID. The subscriber state variables are stored in subscriber database on PDS where other subscriber details are available. These variables hold a value as long as the associated subscriber has at least one active session. Once the last session for the subscriber is terminated, these variables will no longer have a value and will no longer be viable for use in policies.
- **Subscriber Remote State Variables:** Persist remotely in a Subscriber Profile Repository (SPR) as long as the subscriber exists in the SPR. Using these variables requires that an SPR/HSS be configured that is capable of storing these variables.

4.101.1 Local Subscriber State Variable

The Policy blockly for each of the services (SM Service, AM Service, UE Policy Service, and PCRF Core) includes **State Variable** section under **Public** Category.

Figure 4-168 State Variables in Policy



- Save: saves a variable
- Load: load a variable
- Remove: Remove a variable
- Remove All: Applicable only for JSON type of variable. This function removes all the variables from the specified context.

You can select the required action for each context.

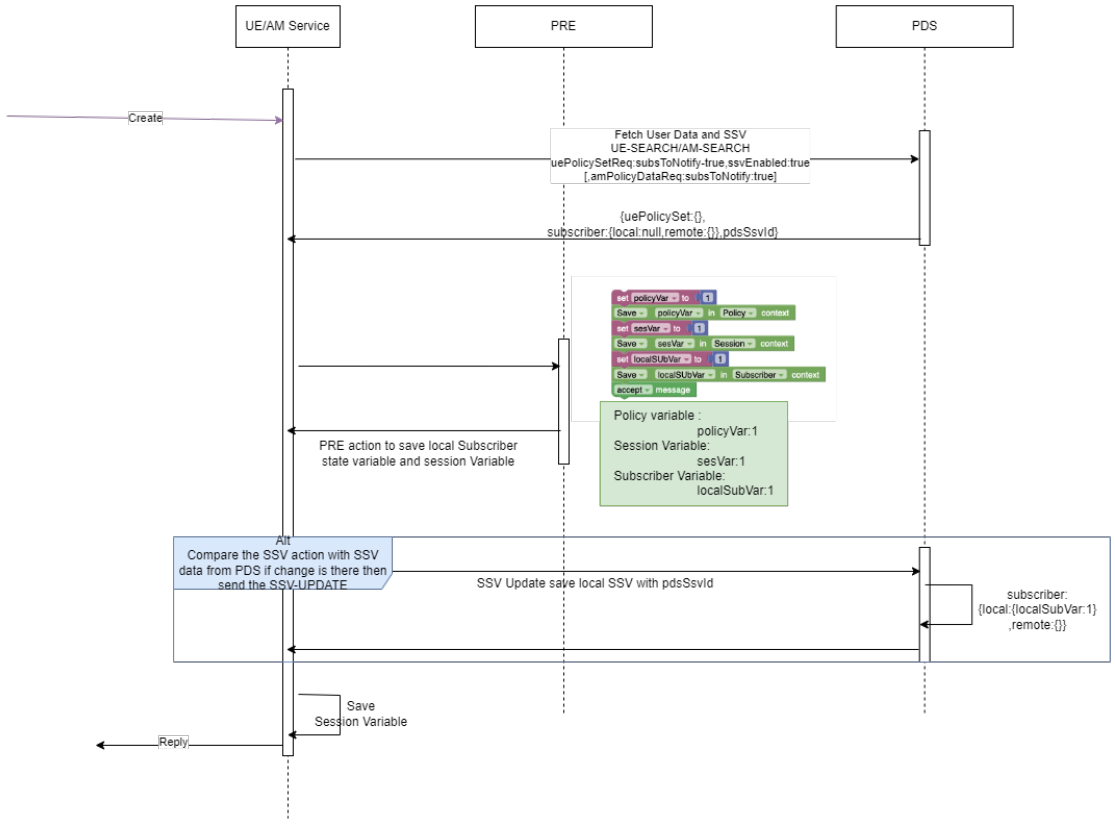
For example, you can select the context as **Policy** and select **Save** to save the specified state variable var and the value assigned to it.

> **Note**
>
> - The context list includes the Remote option to create Subscriber Remote State Variables only for SM Service and UE Policy Service. This option is not available for AM Service.
> - Remove SSV is not revalidated when Query on Update or Query on Delete is enabled in the core service (SM Service or UE Policy Service).

For more information on configuring state variables using blockly, see *State Variables* section in *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.

Call Flow

Figure 4-169 State variables in create session call flow

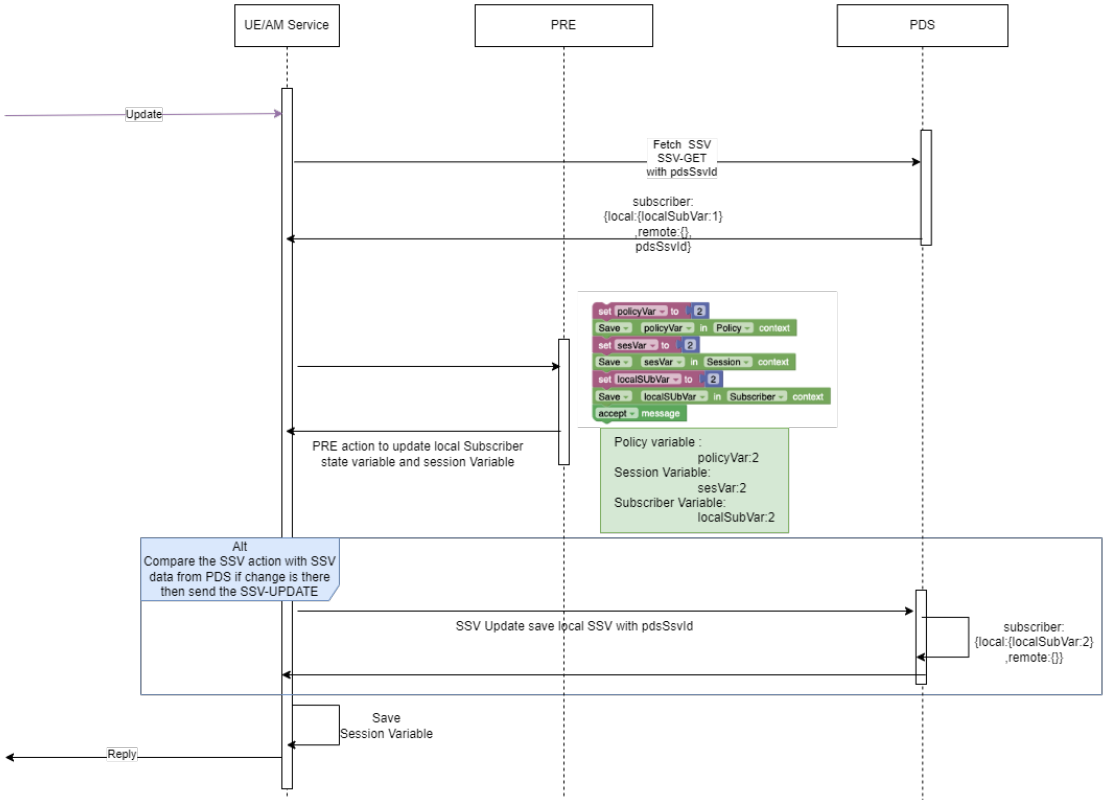


1. Core service (AM Service /UE Policy Service) receives a session create request.
2. If `ssvEnabled` is set to `true`, the core service sends a request to PDS to fetch the Subscriber State Variable (SSV) and the user data.
3. PDS creates a new entry for the SSV in its database. It searches for the user data in PolicyDS database. If it is not present in PolicyDS, it sends a request to the data source such as UDR or CHF through UDR Connector or CHF Connector to get the required details. PDS then responds to the core service with the required information.
4. Core service sends a request to PRE to evaluate the data. PRE creates the Policy state variables as required. PRE does not share these variables with the core service.
5. PRE responds to the core service along with the details of the action to be taken.
6. The core service identifies the session state variables and saves them in the respective session database. Then, it sends the subscriber state variables to PDS to save in the subscriber database.

> **Note**
>
> The core service sends the subscriber state variables to PDS only when there is an update to the variables.

7. After the core service receives a response from PDS and also the core service saves the Session State variables in session database, it replies to SMF/AMF.

Figure 4-170 State variables in update session call flow

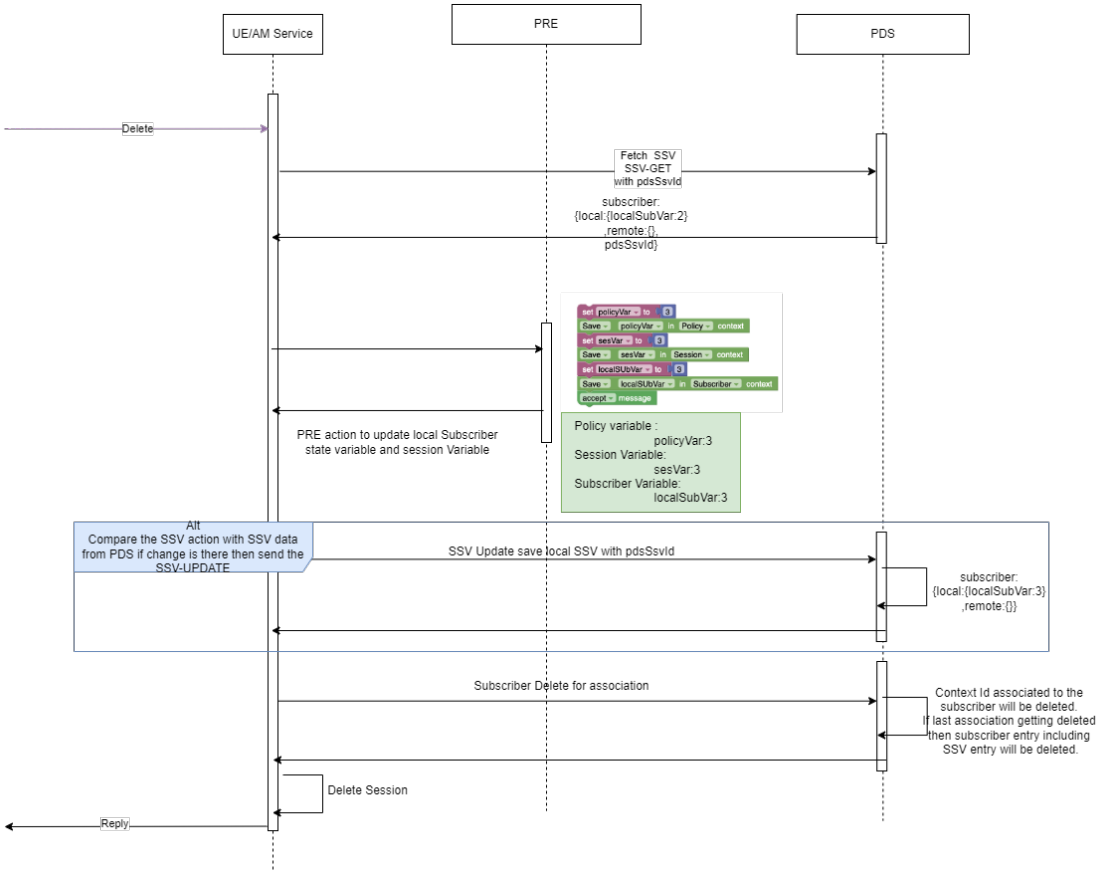


1. Core service (AM Service /UE Policy Service) receives a session update request.
2. If `ssvEnabled` is set to `true`, the core service sends a request to PDS to fetch the Subscriber state Variable (SSV).
3. PDS responds to the core service with the required information.
4. Core service sends a request to PRE to evaluate the data.
5. PRE responds to the core service along with the details of the action to be taken.
6. The core service identifies the session state variables and saves them in the respective session database. Then, it sends the subscriber state variables to PDS to save in the subscriber database.

   Note: The core service sends the subscriber state variables to PDS only when there is an update to the variables.
7. After the core service receives a response from PDS, and also saves the State Session Variables in its session database, it replies to SMF/AMF.

> **Note**
>
> In case of error, session retry is not yet supported.

Figure 4-171 State variables in delete session call flow

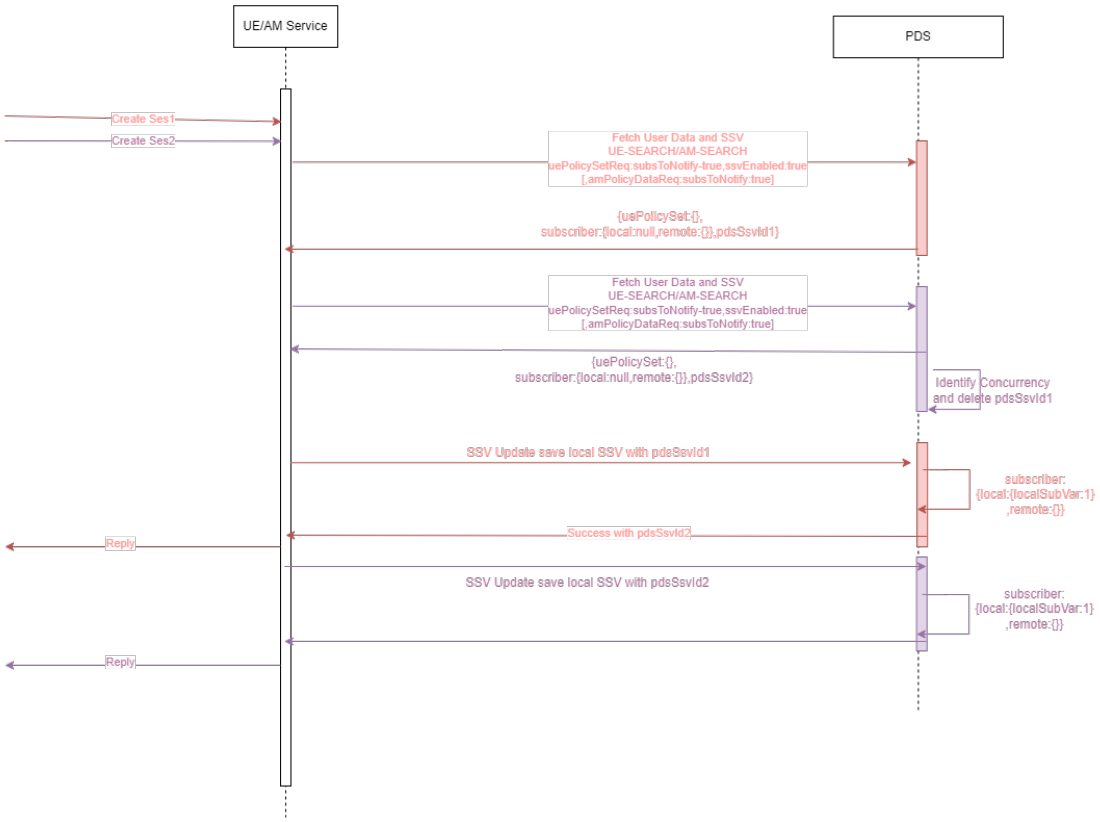


1. Core service (AM Service /UE Policy Service) receives a session delete request.
2. If `ssvEnabled` is set to `true`, the core service sends a request to PDS to fetch the Subscriber state Variable (SSV).
3. PDS responds to the core service with the required information.
4. Core service sends a request to PRE to evaluate the data.
5. PRE responds to the core service along with the details of the action to be taken.
6. The core service identifies the session state variables and saves them in the respective session database. Then, it sends the subscriber state variables to PDS to save in the subscriber database.

   Note: The core service sends the subscriber state variables to PDS only when there is an update to the variables.
7. After the core service receives a response from PDS, and also saves the State Session Variables in its session database

8. Core service sends the subscriber delete request to PDS to delete the subscriber delete for the association.
9. After the core service receives a response from PDS, it deletes the session details from its session database and responds to AMF/SMF.

Figure 4-172 State variables in concurrent flows



1. Core service (AM Service /UE Policy Service) receives multiple create session requests simultaneously.
2. If `ssvEnabled` is set to `true`, the core service sends two separate requests to PDS to fetch the Subscriber state Variable (SSV) for both the requests.
3. PDS creates the entries for SSVs and profiles for both the requests and responds to the core service with the required information.
4. When PDS identifies the concurrency either during UDR subscription call flow or during the subsequent create request, PDS deletes the Subscriber State Variable for the first request and merges the rows in PDS database. When PDS receives the update session for the same SSVID as the first create request, the update request includes the SSVID of the first request along with the PDS SSVID.
5. PDSreturns the update to the core service indicating that it is a new SSV ID and the previous one is deleted.
6. The core service updates the same SSVID in its database.

Managing Subscriber Variables in Policy

Enable

You can enable the Subscriber Variables for SM Service, AM Service, UE Policy Service, and PCRF Core using CNC Console.

- To enable the Subscriber Variables for SM Service, configure **Disable Subscriber Variables** paramter under **User** section in **PCF Session Management** page. For more details, see PCF Session Management.
- To enable the Subscriber Variables for AM Service, configure **Enable Subscriber State Variables** paramter under **User** section in **PCF Access and Mobility** page. For more details, see PCF Access and Mobility.
- To enable the Subscriber Variables for UE Policy Service, configure **Enable Subscriber State Variables** paramter under **User** section in **PCF UE Policy** page. For more details, see PCF UE Policy Service.
- To enable the Subscriber Variables for PCRF Core Service, configure **Enable Subscriber Variables** paramter under **User** section in **Settings** page in **PCRF Core**. For more details, see PCRF Core Service Configurations.

Configure

You can configure the updates to PDS SSV entry's context information for AM context owner and UE context owner using the following flags in `values.yaml` file.

- `resetContextSsvOnAMCreate`
- `resetContextSSVOnUECreate`
- `enableSsvIdForReqParam`

For more information, see *Enabling/Disabling Services Configurations* section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*.

Configure using CNC Console

You can configure the state variables for Policy using CNC Console.

- For details on configuring state variables for SM Service, see PCF Session Management.
- For details on configuring state variables for AM Service, see PCF Access and Mobility.
- For details on configuring state variables for U Policy Service, see PCF UE Policy Service.
- For details on configuring state variables for U Policy Service, see PCRF Core Service Configurations.

Observe

The following PolicyDS metrics provide information about state variables:

- client_request_total
- client_response_total

For more information, see Policy DS Metrics.

4.101.2 Remote Subscriber State Variable

Policy provides UE Policies, via the AMF transparently to the UE. The UE Policy is divided into Policy sections. Such Policy sections are predefined in PCF. PCF either retrieves from UDR, or dynamically generates them. The PCF interfaces with User Data Repository (UDR) to receive subscriber-related Policies for User Equipment (UE). UDR stores the subscriber attributes. PCF communicates with UDR to receive these attributes which are used in the evaluation of Policies.

PCF support for dynamic data for UE Policies:

Subscriber State Variable (SSV) comprises of local and remote section. The remote SSV section is formed by extracting the data from UDR response as per the UEPolicySet datasource path, configured at PDS Settings in the CNC Console. SSV section is sent back to UE service as part of create request processing. UE service forwards the SSV to Policy Runtime Engine(PRE) for:

- Evaluation of Policies, and
- Update the changes in the state variable and send it to UDR.

Revalidation of dynamic data for UE Policies:

UE service supports 'ResetContext' and 'Revalidation' feature. On enabling this feature, an updated SSV remote Policy section is formed with revalidated UE data. This section is updated in the database and sent to UE service.

Write Remote Subscriber State Variable in UDR as a PATCH request:

UE service sends SSV to PRE service for evaluation of Policies. The policy evaluation results in addition or modification or deletion in the SSV. The PRE sends back the updated SSV to UE service. UE service sends the updated SSV to Policy Data Service (PDS). PDS service sends the remote SSV section toward UDR as PATCH request through UDR Connector. The PATCH request that PDS sends to UDR Connector will also include the data source information. UDR Connector uses this data source information to forward the PATCH to appropriate UDR. If the data source information is not included in the PATCH, it will be determined by the UDR-Connector.

> **Note**
>
> PCF supports dynamic data for UE policies in independent deployment only. It does not work if SM and UE services are running in the same PCF instance with SSV enabled.

4.102 AMF Selection for Namf-comm Subscription

Policy supports discovering AMF as a producer based on AMF SetID and AMF Region ID, which are extracted from Globally Unique AMF Identifier (GUAMI). This aids in transfering the UE Policy to UE through AMF and sending initial subscription request to AMF during UE Policy Transfer request. For more details, see UE Policy Enhancements.

4.103 Support for MCPTT Features

Oracle Communications Cloud Native Core, Policy (Policy) supports Mission Critical Push-to-Talk (MCPTT) and other mission-critical data and video over cellular networks. There are many MCPTT applications that are available through cellular Long Term Evolution (LTE), with mobile broadband meeting many emergency responders' needs for reliable coverage, secure communication and real-time video sharing.

This feature enables PCRF support for MCPTT related QoS Class Identifier (QCI)s, and MCPTT specific features such as Priority sharing and Preemption.

PCRF enables MCPTT for a session based on the supported features exchanged between the Policy and Charging Enforcement Function (PCEF)/Application Function (AF) and PCRF (dynamic discovery of supported features).

Policy suports the following MCPTT features:

- **Allocation Retentions Priority (ARP) Selection**: ARPs determine the priority level, the pre-emption capability and the pre-emption vulnerability of each QoS flow according to the operators' policy. PCRF uses ARPs to determine the relative importance of a resource request and decides whether a new QoS Flow can be accepted or rejected in the case of resource limitations.
- **QoS Class Identifier (QCI) selection**: QCIs are used to ensure carrier traffic is allocated appropriate Quality of Service (QoS). Different carrier traffic requires different QoS and therefore different QCI values.

  The following are the MCPTT related QCIs:

  – QCI-65 - MCPTT Audio service flow.
  – QCI-66 - Non-MCPTT Audio service flow.
  – QCI-67 (for MC Video)
  – QCI-69 - MCPTT signaling flow.
  – QCI-70 - MCPTT Data service flow.
  – QCI-7 - Voice, video (live streaming)
  – QCI_128-254 - Operator specific
- **Priority Sharing**: For all the Policy and Charging Control Rules (PCC rules) with the same QCI assigned and having an associated priority sharing indicator, PCRF assigns the same Allocation and Retention Priority (ARP) settings. Having the same setting for the ARP enables the usage of the same bearer.

  PCRF decides the ARP for the PCC rules (same QCI and Priority Sharing indicated by AF) as follows:

  – ARP Priority is set as highest of the priority among all the PCC rules
  – ARP pre-emption capability is set if any of the original PCC rules have the ARP pre-emption capability set
  – ARP pre-emption vulnerability is set if all the original PCC rules have the ARP pre-emption vulnerability set
- **Preemption control**: If PCEF informs the PCRF that a PCC rule provisioning or modification failed (due to resource reservation failure), PCRF can apply pre-emption and remove the active PCC rules from PCEF and retry the PCC rule provisioning or modification procedure.

  If PCRF does not apply pre-emption, AF is notified that the resource reservation for the new media flow failed.

Feature enablement is controlled by the supported feature negotiation done between the PCRF and the PCEF/AF. By default, all MCPTT related feature bits are enabled and applied to session (when supported by PCEF and AF).

Table 4-67 MCPTT related features in Gx

Feature Name	Feature bit	M/O	Feature list Id	Vendor Id
MissionCriticalQCIs	25	O	1	10415
MCVideoQCI	9	O	2	10415
MCVideo	22	O	1	10415

Table 4-68 MCPTT related features in Rx

Feature Name	Feature bit	M/O	Feature list Id	Vendor Id
MCPTT	17	O	1	10415
PrioritySharing	18	O	1	10415
MCPTT-Preemption	21	O	1	10415

You must mention the supported features separated by comma as values in these keys.

This is not limited to MCPTT feature bits. You can disable any of the existing optional features using this option.

Mandatory features such as Rel8, Rel9 and Rel10 are enabled by default for negotiation.

Selecting QCIs

You can select the required QCIs using the existing Policy actions.

You must assign these QCIs to the PCC rules using Policy and PCC rule configurations. When there is no decision received from Policy suggesting QCI for MCPTT session, QCIs are chosen based on the media type.

For example:

Table 4-69 QCI based on Media Type

Media Type	QCI
Audio	QCI_65
Data	QCI_70
*Anything else	QCI_69

Selecting ARPs

You can configure the ARP parameters such as priority, Preemption capability and preemption vulnerability for default EPS bearer and PCC rules generated for the AF flows.

When there is no ARP set from Policy, the ARP settings for MCPTT session is created under PCRF core service configuration.

> **Note**
>
> ARP assigned based on Policy is considered as final. This can be overridden by priority sharing feature (when enabled). But the original ARP decision is stored for future use.

Priority Sharing

When priority sharing feature is enabled, the ARP assigned by policy or configuration canbe overridden (when conditions are met). The original ARP assigned for the PCC Rule is stored along with the overridden ARP as part of the session data stored in the database. The original ARP information is required when there is a need to apply the priority sharing again on these PCC rules in the future.

Table 4-70 Sample priority sharing scenario

PCC Rule 1 original ARP			PCC Rule 2 original ARP	Shared ARP				
Priority	**Preemption Capability**	**Preemption Vulnerability**	**Priority**	**Preemption Capability**	**Preemption Vulnerability**	**Priority**	**Preemption Capability**	**Preemption Vulnerability**
1	Disabled	Disabled	2	Enabled	Enabled	1 PCC Rule 1 - unchanged PCC Rule 2 - changed	Enabled PCC Rule 1 - changed PCC Rule 2 - unchanged	Disabled PCC Rule 1 - unchanged PCC Rule 2 - changed

In this case, both the PCC rules belong to the same IPCAN session, same QCI and having priority sharing indicator.

Following are the scenarios when Priority sharing rederive ARP is triggered:

- At least one of the PCC Rules (sharing same ARP) is removed
- A new PCC Rule (with same QCI, Priority sharing indicator) is created
- The original derived ARP of at least one of the PCC rule changes

Whenever the ARP of an already installed PCC rule changes, that rule will also be added as part of the Charging-Rule-Install Avp. For the rules to be modified, the charging-Rule-Definition will have the new ARP values.

Since priority sharing is applied at different scenarios, the original ARP and the priority sharing indication (received from AF) becomes necessary, These are mentioned as values of *PCRFDiameterEnfAppFlow* paremeter stored in the Gx Session (in the database).

Preemption

Whenever a PCC rule (with priority sharing enabled) installation or modification fails due to resource reservation issue, any other preemptable PCC Rule(s) is removed and the failed PCC Rule will be retried.

Preemption is triggered only when the following conditions are met

- When PCC rules (with priority sharing) provisioning or modification fails (Charging-Rule-Report with rule failure code `RESOURCE_ALLOCATION_FAILURE`)
- Failed PCC Rule(s) preemption capability (original - before Priority sharing) is `ENABLED`
- At least one PCC rule with priority equal or lower (than the failed PCC rule) and having preemption vulnerability `ENABLED`
- MCPTT-Preemption is enabled (supported features)

Preemption is not triggered if any of the above mentioned conditions are not met. Instead, a Re-Authorization Request (RAR) is triggered towards AF (if requested by AF) with the Specific-Action AVP set to `INDICATION_OF_FAILED_RESOURCES_ALLOCATION` to report the resource allocation failure.

When there are more than one potential PCC rule candidate for preemption, the PCC rule to be preempted is selected based on the Pre-emption-Control-Info AVP received from Authorization Authentication Request (AAR) or the local configuration (in PCRF-core service configuration).

The control information has the following three options.

- Most recent added flow - Newest flow among the potential preemptable rules to be preempted
- Least recent added flow - Oldest flow among the potential preemptable rules to be preempted
- Highest bandwidth flow - Having highest bandwidth among the potential preemptable rules

The rule to be preempted is selected based on the information received from AF or local configuration.

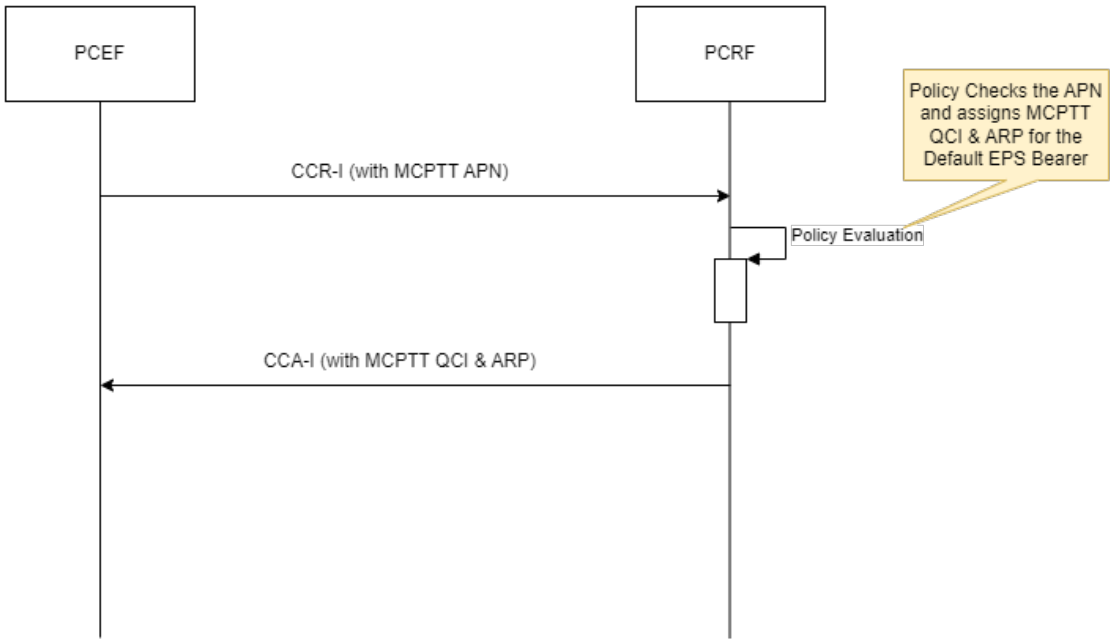
The rule creation time stored in the PCC rule information (part of session object) is used to compare the oldest and the newest rule,

After the Preemption, Priority sharing is triggered again to rederive the ARP for the affected PCC Rules (sharing the same QCI of the preempted PCC Rule).

> **Note**
>
> Currently, Policy does not support preemption using Policy. Preemption is performed as per the procedure defined by 3GPP only.
>
> For the preempted PCC rules a Rx-RAR will be sent to AF to notify about pre-emption of the flow (with specific action set to `INDICATION_OF_RELEASE_OF_BEARER`)

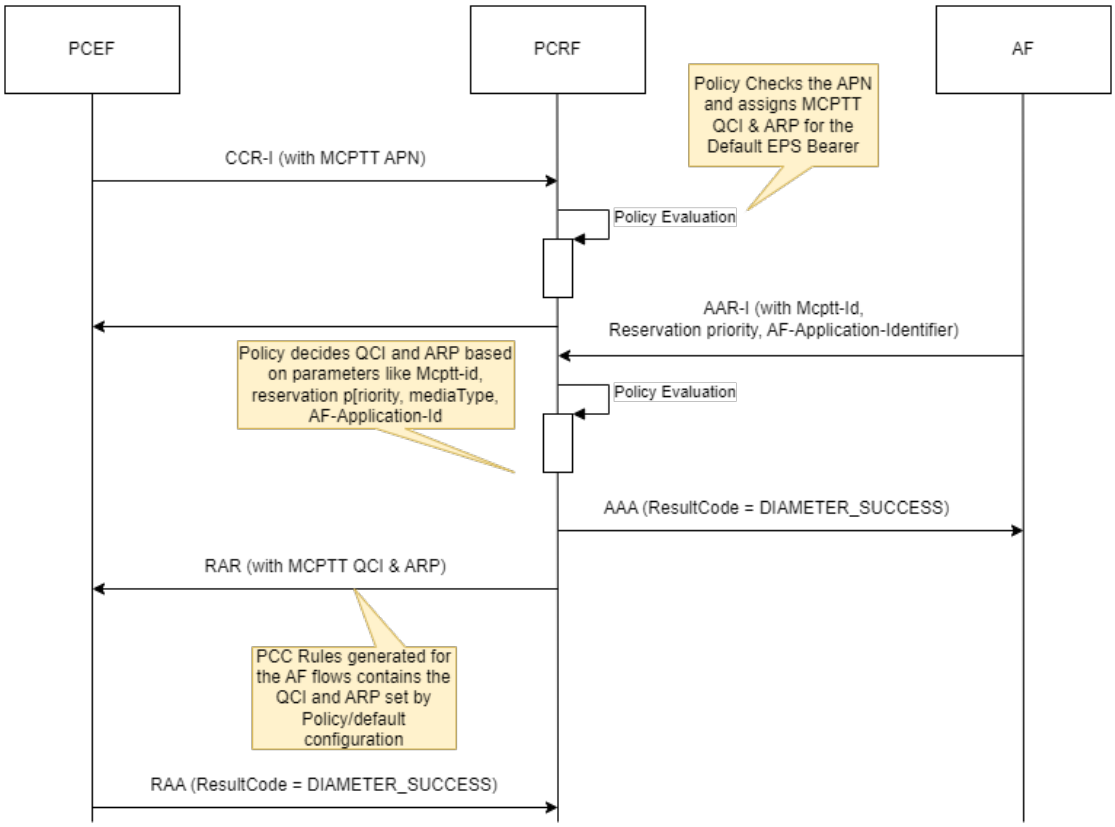
Call Flow

Figure 4-173 MCPTT APN Handling



1. The PCEF sends a Credit Control Request-Initial (CCR-I) message to the PCRF with MCPTT APN details.
2. PCRF checks the Access Point Name (APN) and assigns the MCPTT QCI and ARP for the default Evolved Packet System (EPS) Bearer.
3. PCRF sends a Credit Control Answer-initial (CCA-I) message to PCEF and provisions the MCPTT QCI and the ARP settings.

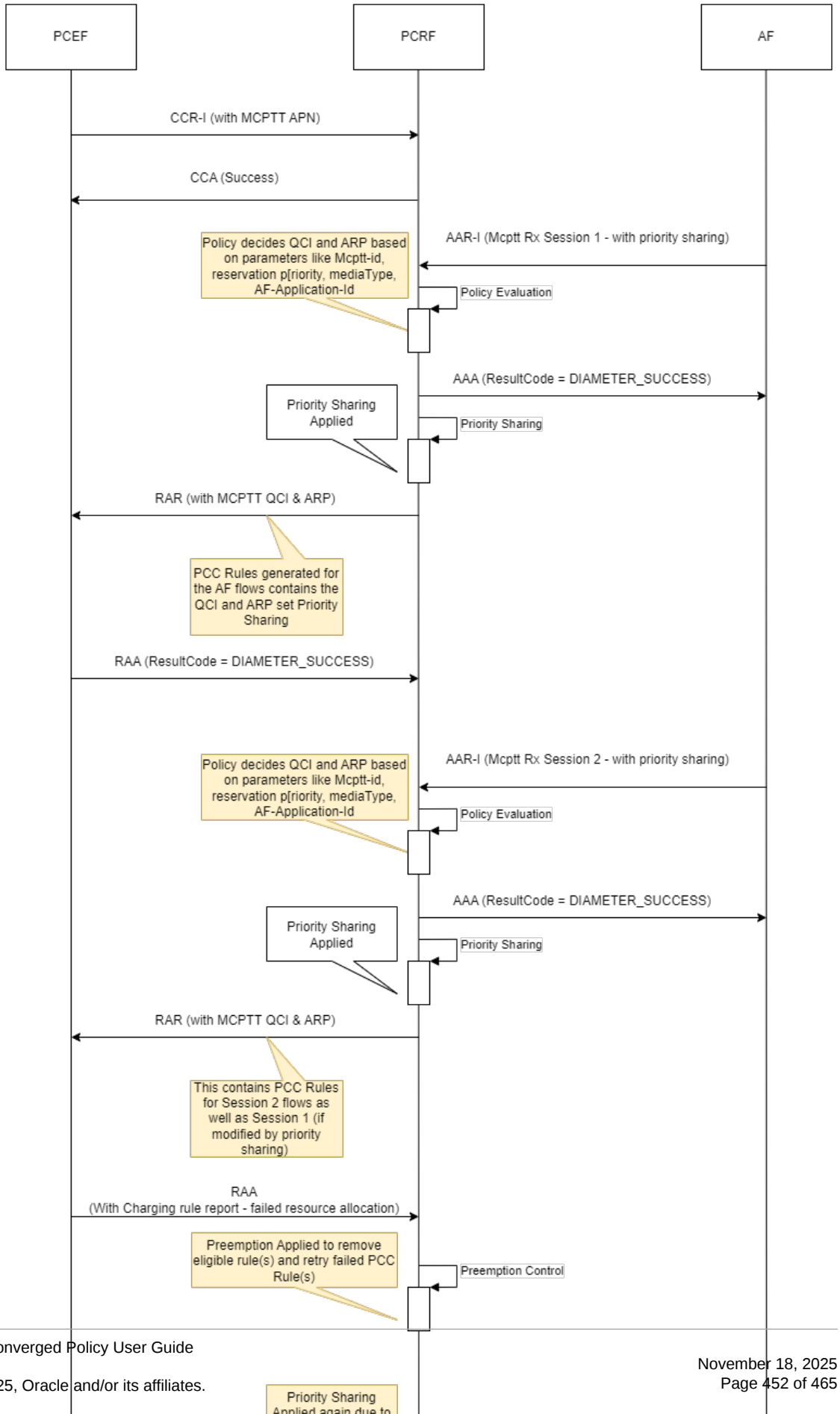
Figure 4-174 MCPTT QCI & ARP settings for MCPTT AF flows



1. The PCEF sends a Cedit Control \Rrequest initial (CCR-I) message to the PCRF with MCPTT APN details.
2. PCRF checks the APN and assigns the MCPTT QCI and ARP for the default EPS Bearer.
3. PCRF sends a credit Control Answer initial (CCA-I) message to PCEF and provisions the MCPTT QCI and the ARP settings.
4. PCRF receives an AAR message from AF with the session information, which includes MCPTT-ID, Reservation Priority, and the AF-Application-identifier associated with the session.
5. PCRF decides the QCI and the ARP based on the parameters received from AF.
6. The PCRF sends a Diameter Authentication, Authorization, and Accounting (AAA) to the AF with `ResultCode=DIAMETER_SUCCESS`.
7. PCRF sends a Reauthorization Request (RAR) message to PCEF to provision the QCI and the ARP. The PCC Rules generated for the AF flows contains the QCI and ARP set by Policy/default configuration.

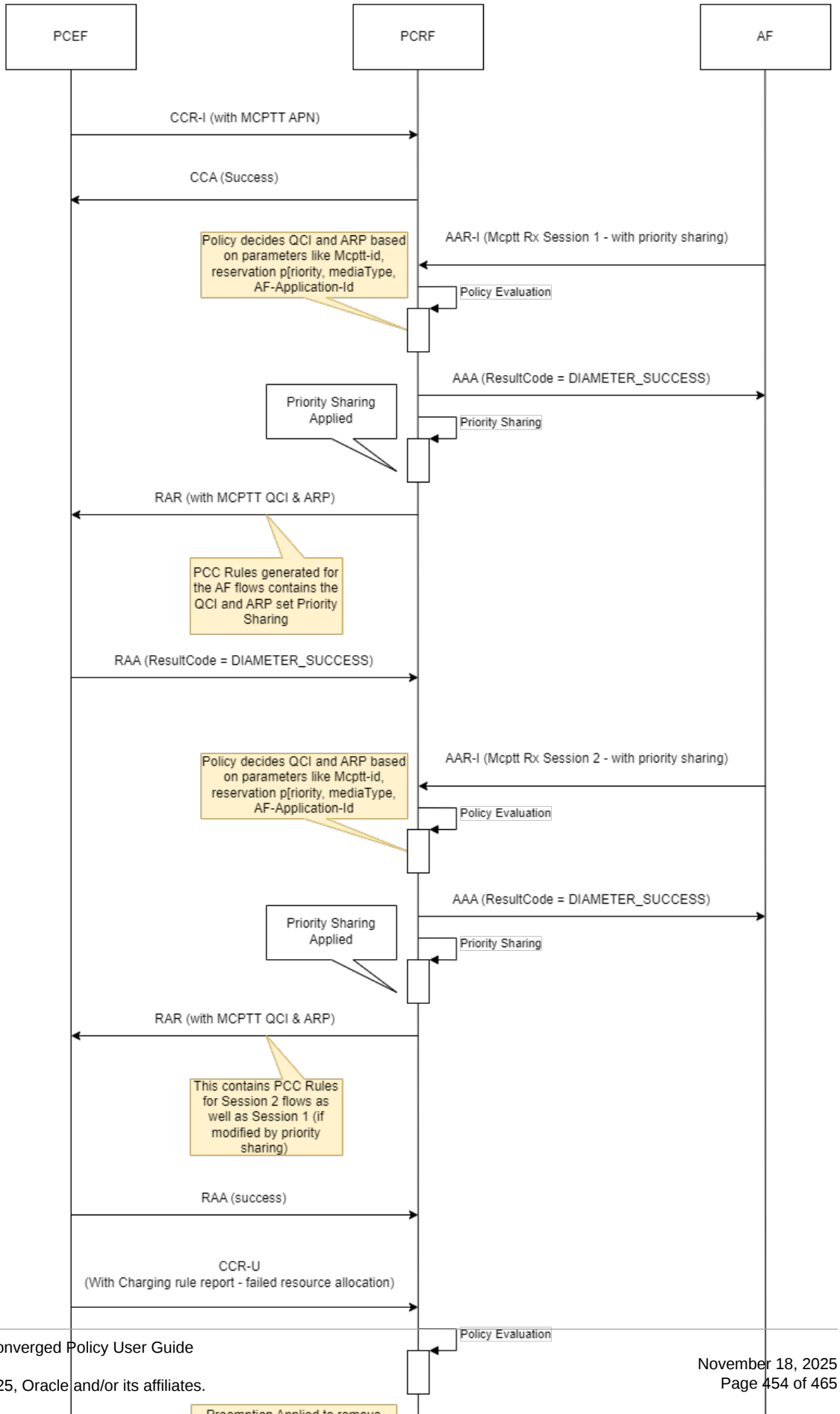
8. PCEF responds to PCRF with a Reauthorization answer (RAA) message with `ResultCode=DIAMETER_SUCCESS`).

Figure 4-175 Priority Sharing



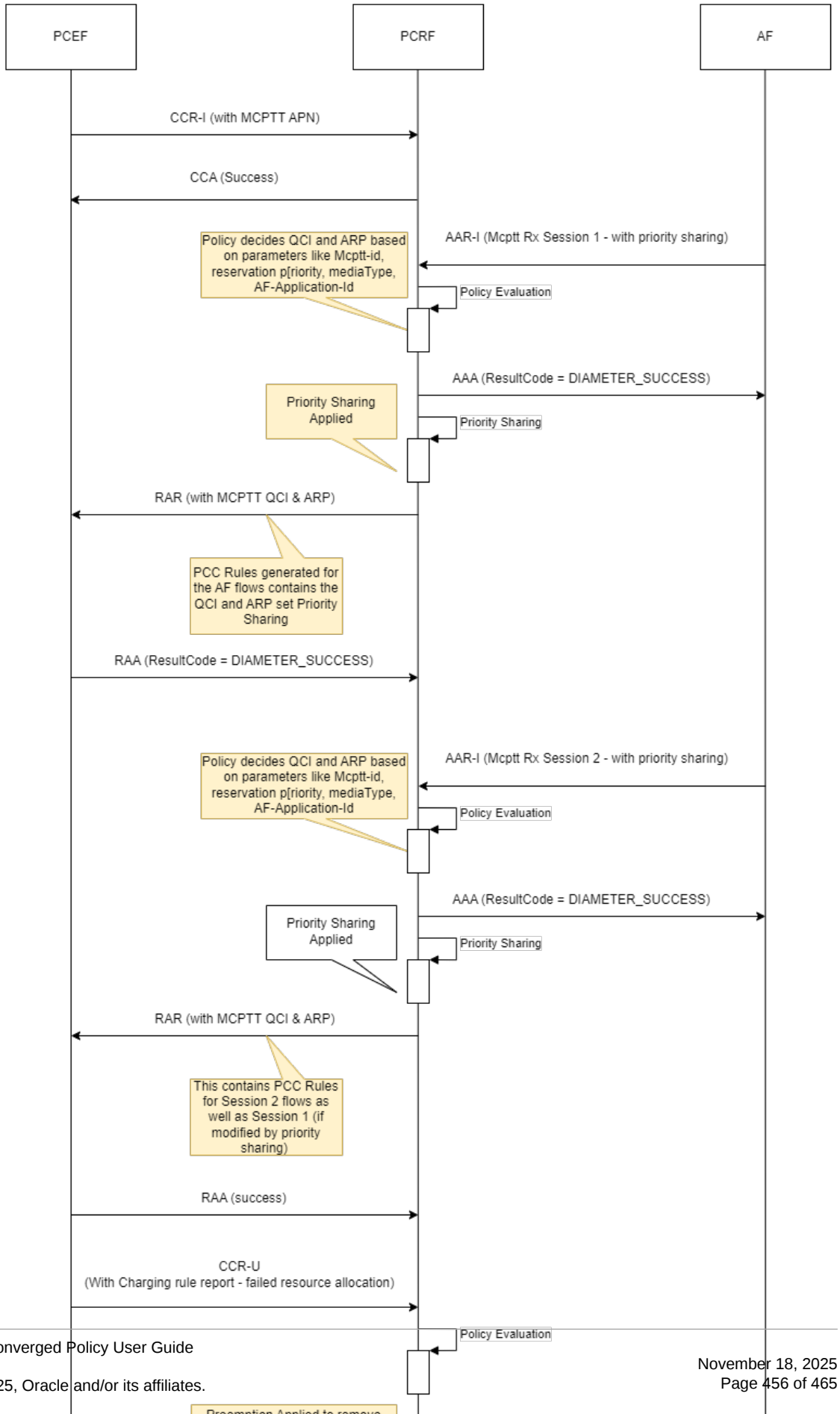
1. The PCEF sends a credit control request initial (CCR-I) message to the PCRF with MCPTT APN details.
2. PCRF sends a credit control answer initial (CCA-I) success message to PCEF.
3. PCRF receives an AA-Request Initial (AAR-I) messages from AF for Rx Session 1 with priority sharing enabled.
4. PCRF decides the QCI and the ARP based on the parameters received from AF.
5. The PCRF sends a Diameter AAA to the AF with `ResultCode=DIAMETER_SUCCESS`.
6. PCRF applies Priority sharing.
7. PCRF sends a Reauthorization Request (RAR) message to PCEF to provision the QCI and the ARP. The PCC Rules for session 1 generated for the AF flows contains the QCI and ARP set by Policy/default configuration.
8. PCEF responds to PCRF with a Reauthorization answer (RAA) message with `ResultCode=DIAMETER_SUCCESS`).
9. PCRF receives an AA-Request (AAR) messages from AF for Rx Session 2 with priority sharing enabled.
10. PCRF decides the QCI and the ARP based on the parameters received from AF.
11. The PCRF sends a Diameter AAA to the AF with `ResultCode=DIAMETER_SUCCESS`.
12. PCRF applies priority sharing.
13. PCRF sends a Reauthorization Request (RAR) message to PCEF to provision the QCI and the ARP. The PCC Rules for session 2 as well as session 1 generated for the AF flows contains the QCI and ARP set by Policy/default configuration.
14. PCEF responds to PCRF with a Reauthorization answer (RAA) message with `ResultCode=DIAMETER_SUCCESS`).
15. PCRF receives a Session-Termination-Request (STR) from AF.
16. AF responds to PCRF with a Session-Termination-Answer (STA).
17. PCRF applies priority sharing.
18. PCRF sends an RAR to remove and update the PCC rules. The RAR message contains the PCC rules removed for Session 2 flows as well as session 1 (if modified by priority sharing).
19. PCEF responds to PCRF with a Reauthorization answer (RAA) message with `ResultCode=DIAMETER_SUCCESS`).

Figure 4-176 Preemption Control - CCRU- ChargingRuleReport scenario



1. The PCEF sends a credit control request initial (CCR-I) message to the PCRF with MCPTT APN details.
2. PCRF sends a credit control answer initial (CCA-I) success message to PCEF.
3. PCRF receives an AA-Request initial (AAR-I) messages from AF for Rx Session 1 with priority sharing and preemption enabled.
4. PCRF decides the QCI and the ARP based on the parameters received from AF.
5. The PCRF sends a Authentication, Authorization, Accounting (AAA) message to the AF with `ResultCode=DIAMETER_SUCCESS`.
6. PCRF applies Priority sharing.
7. PCRF sends a Reauthorization Request (RAR) message to PCEF to provision the QCI and the ARP. The PCC Rules for session 1 generated for the AF flows contains the QCI and ARP set by Policy/default configuration.
8. PCEF responds to PCRF with a Reauthorization answer (RAA) message with ResultCode=DIAMETER_SUCCESS).
9. PCRF receives an AA-Request (AAR) messages from AF for Rx Session 2 with priority sharing and preemption enabled.
10. PCRF decides the QCI and the ARP based on the parameters received from AF.
11. The PCRF sends a Diameter AAA to the AF with `ResultCode=DIAMETER_SUCCESS`.
12. PCRF applies priority sharing.
13. PCRF sends a Reauthorization Request (RAR) message to PCEF to provision the QCI and the ARP. The PCC Rules for session 2 as well as session 1 generated for the AF flows contains the QCI and ARP set by Policy/default configuration.
14. PCEF responds to PCRF with a Reauthorization answer (RAA) message with Charging rule report - failed resource allocation.
15. Also, PCEF sends a Credit Control Request - Update (CCR-U) with charging rule report indicating failed resource allocation.
16. PCRF evaluates the details, and applies priority control and tries the preemption once again to remove eligible rules and retry failed PCC rules.
17. PCRF applies priority sharing again due to removed PCC rules.
18. PCRF sends a Credit Control Answer-Update (CCA-U) to PCEF to remove and update the PCC Rules.
19. PCRF sends an RAR or ASR to AF informing the failed resource allocation for the preempted flow.
20. AF responds to PCRF with a Reauthorization answer (RAA) message.

Figure 4-177 Preemption - No Preemptable Rule found



1. The PCEF sends a credit control request initial (CCR-I) message to the PCRF with MCPTT APN details.
2. PCRF sends a credit control answer initial (CCA-I) success message to PCEF.
3. PCRF receives an AA-Request initial (AAR-I) messages from AF for Rx Session 1 with priority sharing enabled.
4. PCRF decides the QCI and the ARP based on the parameters received from AF.
5. The PCRF sends a Authentication, Authorization, Accounting (AAA) message to the AF with `ResultCode=DIAMETER_SUCCESS`.
6. PCRF applies Priority sharing.
7. PCRF sends a Reauthorization Request (RAR) message to PCEF to provision the QCI and the ARP. The PCC Rules for session 1 generated for the AF flows contains the QCI and ARP set by Policy/default configuration.
8. PCEF responds to PCRF with a Reauthorization answer (RAA) message with `ResultCode=DIAMETER_SUCCESS`).
9. PCRF receives an AA-Request (AAR) messages from AF for Rx Session 2 with priority sharing enabled.
10. PCRF decides the QCI and the ARP based on the parameters received from AF.
11. The PCRF sends a Diameter AAA to the AF with `ResultCode=DIAMETER_SUCCESS`.
12. PCRF applies priority sharing.
13. PCRF sends a Reauthorization Request (RAR) message to PCEF to provision the QCI and the ARP. The PCC Rules for session 2 as well as session 1 generated for the AF flows contains the QCI and ARP set by Policy/default configuration.
14. PCEF responds to PCRF with a Reauthorization answer (RAA) success message.
15. Also, PCEF sends a Credit Control Request - Update (CCR-U) with charging rule report indicating failed resource allocation.
16. PCRF evaluates the details, and applies priority control and tries the preemption once again to remove eligible rules and retry failed PCC rules.
17. PCRF sends a RAR message to AF informing the failed resource allocation for the preempted flow.
18. PCRF sends a Credit Control Answer-Update (CCA-U) to PCEF indicating no change in the PCC Rules.
19. AF responds to PCRF with a Reauthorization answer (RAA) message.

Managing Support for MCPTT features

Enable

By default, the MCPTT features are enabled.

You can disable the required MCPTT related feature bits using the following two keys under **Advance Settings** in **Settings** page for **PCRF Core** on Oracle Communications Cloud Native Core, Cloud Native Configuration Console (CNC Console):

- Gx.DisableSupportedFeatures (for Gx)
- Rx.DisableSupportedFeatures (for Rx)

For more details on enabling/disabling the MCPTT features using CNC Console, see Settings.

Configure

You can configure the support for MCPTT features using CNC Console and REST API.

Configure using CNC Console

- You can configure the default ARP and default preemption control by applying the configurations under **MCPTT** section in **Settings** page for **PCRF Core** on Oracle Communications Cloud Native Core, Cloud Native Configuration Console (CNC Console). For more details, see Settings.
- To assign the QCIs, apply the configurations for **Use Custom QoS Class Identifier** and **QoS Class Identifier** parameters while creating the PCC rule on **Create PCC Rule** page. For more information, see PCC Rule.
- To configure PCC profile, apply the configurations for **Use Custom QoS Class Identifier** and **QoS Class Identifier** parameters while creating the PCC rule profile on **Create PCC Rule Profile** page. For more information, see PCC Rule.
- You can use the blockly conditions and actions to set QCI and ARP default EPS bearer. For more information, see *Priority/Emergency* section in *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.

Configure using REST API

You can configure PCC Rule, PCC Profile, PCRF-core using the following REST APIs:

- `Core Service`
- `Traffic for PCC Rule`
- `Traffic for PCC Rule Profile`

For more information on configuring using REST API, see *Policy REST Specifications* section in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

4.104 Support for Listing and Comparing UPSIs Received from UDR

Converged Policy provides blocks, conditions and actions to support finding the delta between the UPSI's id list:

- currently configured in PCF
- UPSI's that are send on UE Policy Registration
- UPSIs that are on UDR

It allows the user to install/remove this delta or a subset of delta.

Also, it allows to access the attributes of the UPSI object (UPSC, MCC and MNC) and can be able to check the URSPs that are configured to be installed under a specific UPSI.

For details on the UE Policy blocks, actions, and conditions used for this functionality, see *UE Policy Bloaks* section in *Oracle Communications Cloud Native Core, Converged Policy Design Guide.*

4.105 Roaming Support in Policy

PCRF supports S8 Home Routing scenario for LTE and VoLTE networks in 4G deployment.

S8 Home Routing allows the home operator to control the services and policies applied to the roaming subscriber. S8 interface connects the visited network's Serving Gateway (SGW) to the home network's PGW.

Since the data is routed through the home network's PGW, the home network's PCRF applies the subscriber's policy and charging rules such as QoS and PCC Rules.

The PCRF at home network interfaces the home network PGW and applies the policies on Gx interface, such as identifying a roaming subscriber from 3GPP-SGSN-MCC-MNC AVP and applying differentiated QoS parameters for the roaming subscribers.

For details on how to write a policy based on MCC and MNC, see *Creating Policies* section in *Cloud Native Core, Converged Policy Design Guide*.

> **Note**
>
> Currently, Policy do not support:
>
> - Roaming in 5G deployment (using PCF).
> - Local Breakout (LBO)

4.106 Wi-Fi Support using RAT-Type AVPs

To support Wi-Fi (WLAN) and voice over Wi-Fi, RAT-type Attribute-Value Pair (AVP)s are included in Credit-Control-Request (CCR) messages to inform the CnPCRF about the access technology through which the UE is connected. RAT-Type is usually included as an enumerated AVP, where each type of RAT is represented by a predefined integer value.

The following are some of the values allowed for RAT-Type AVPs:

- EUTRAN (LTE)
- WLAN (Wi-Fi)
- UTRAN (3G)

Based on the RAT-Type, the CnPCRF can apply different Policy rules.

For example, different quality of service (QoS) and charging rules can be applied for a user connected using Wi-Fi (WLAN) compared to a user connected through LTE (EUTRAN).

When a UE moves from one RAT to another (for example, from LTE to Wi-Fi), the network triggers a CCR-U message with the updated RAT-Type AVP. This allows the PCRF to dynamically adjust policies in real-time.

RAT type in conjunction with other parameters in the CCR, such as user location information, subscriber profile from data sources enables operators to create dynamic, and user-centric policies in CnPCRF.

Enable

This feature is enabled automatically at the time of Policy installation.

4.107 Support for Database Slicing

Whenever Policy services (SM service, PCRF Core, or Usage Monitoring service) receives a Create, Retrieve, Update, or Delete request from any of the microservices, the request is processed by the main database (SMPolicyAssociation database, UMContext, GxSession).

With the use of MySQL NDB software, there is a limitation on having a single replication channel for a given database. When the commit rate for a database is very high, the replication of the database impacts the replication performance depending upon the number of commits as well as the commit (record) size.

In order to overcome the replication limitations of the database, especially in a multisite environment, the database can be sliced based on a particular slicing criteria. The database tables are divided into different slices based on the Helm configuration (such as SM_POLICY_ASSOCIATION_TABLE_SLICING_COUNT for SMPolicyAssociaion, UM_CONTEXT_TABLE_SLICING_COUNT for UMContext and GX_SESSION_TABLE_SLICING_COUNT for GxSession). After the slices are created, the database records are distributed across all the slices.

With this database slicing, instead of the main database processing all the requests, some of the database operations are processed using sliced tables.

The number of slices can be configured depending upon the replication requirements.

For example, if a single slice is able to replicate up to ~40K TPS and the immediate TPS need is ~75K, two slices can support the immediate required load. If the estimated projected growth YoY is 25%, the TPS need after 5 years reaches around ~200K TPS. Thus a table slice count of 8 can be used to manage the database load for the next 5 years.

> **Note**
>
> This calculation takes into consideration only replication channel and not any other challenges that may come up for 200K TPS.

By default, there is no separate replication channel for each of these slices. Replication channels can be configured. For example, two replication channels can be configured, each replicating 4 slices. This configuration can change depending on the increase in TPS and available bandwidth across the replicating sites.

Managing the Support for Database Slicing

Slicing SMPolicyAssociation for SM service

While upgrading from an older version of Policy to a later version, you can configure the Database Slicing feature for SM service database (SMPolicyAssociation table) as follows:

1. In CNC Console, under **Service Configurations** on **PCF Session Management** page, set the value of ENABLE_SM_POLICY_ASSOCIATION_TABLE_SLICING advanced settings key to `true`. This will enable the database slicing on SMPolicyAssociation. By default, the value of this key is set to `false`. Once the database slicing is enabled, the database operations are processed on the sliced tables.

   When ENABLE_SM_POLICY_ASSOCIATION_TABLE_SLICING is enabled, the Audit Service audits the SmPolicyAssociation sliced tables as well.

2. Configure the number of slices using SM_POLICY_ASSOCIATION_TABLE_SLICING_COUNT Helm parameter.

   By default, the slicing count is set to 1.

   During the upgrade, add the Helm configuration in custom.yaml file to configure SM_POLICY_ASSOCIATION_TABLE_SLICING_COUNT parameter.

   The slicing table feature is backward compatible, so that the older Create, Retrieve, Update, and Delete operations work as expected. Data is populated in new tables after a successful upgrade and when ENABLE_SM_POLICY_ASSOCIATION_TABLE_SLICING is set `true` using CNC Console.

   > **Note**
   >
   > After enabling the feature, make sure that it is not disabled again because data can not be fetched from sliced tables.

Slicing for UMContext database for Usage Monitoring service

While upgrading from an older version of Policy to a later version, you can use the following Helm paramters to configure the table slicing feature for Usage Monitoring database (umContext table):

- `ENABLE_TABLE_SLICING`: to enable the database slicing feature. By default, this parameter is disabled.
- `UM_CONTEXT_TABLE_SLICING_COUNT`: to configure the number of slices created. By default, the slicing count is set to 1.

The `ENABLE_UM_CONTEXT_TABLE_SLICING` key under **Advanced Settings** in **Usage Monitoring** page for **Service Configurations** in CNC Console is used to enable/disable the database operations on sliced tables. That is, whenever `ENABLE_UM_CONTEXT_TABLE_SLICING` key is set to `true`, the create/retrieve/update/delete operations are performed on the database slices instead of the main database.

By default, `ENABLE_UM_CONTEXT_TABLE_SLICING` key is disabled.

For example, when PCRF Core sends a create request to Usage Monitoring service and `ENABLE_UM_CONTEXT_TABLE_SLICING` flag is enabled, the Usage Monitoring service responds to PCRF Core with **x-oc-slicing-info**, which contains the slicing information of Usage Monitoring database (UmContext table). PCRF Core stores this information in its Gx session database. In subsequent update or terminate requests, PCRF Core uses this in the header.

In case of any change in `UM_CONTEXT_TABLE_SLICING_COUNT` parameter, existing sessions in the UmContext table will not be impacted. Data can still be retrieved from the older slices.

> **Note**
>
> After enabling the `ENABLE_UM_CONTEXT_TABLE_SLICING` flag, make sure not to disable it again. Otherwise, data cannot be fetched from sliced tables.
>
> Also, in case of rollback, data saved on sliced tables cannot accessed. Data saved on the sliced tables must be manually ported to the main database.

In case of a change in `UM_CONTEXT_TABLE_SLICING_COUNT` parameter:

The PREVIOUS_UM_CONTEXT_TABLE_SLICING_COUNT key under **Advanced Settings** in **Usage Monitoring** page for **Service Configurations** in CNC Console is used to store the previous slicing count.

If UM_CONTEXT_TABLE_SLICING_COUNT is changed and the service cannot find the data with the current slicing count, it uses PREVIOUS_UM_CONTEXT_TABLE_SLICING_COUNT to search for the data.

If there are multiple values for PREVIOUS_UM_CONTEXT_TABLE_SLICING_COUNT, the Usage Monitoring service uses the latest previous count and if the data is still not found, it uses the next previous count. That is, if PREVIOUS_UM_CONTEXT_TABLE_SLICING_COUNT contains two values such as 0 and 5, the Usage Monitoring service first uses the previous count as 5 to search for the data. If it is still not found, the service uses the previous count as 0 to search for the data.

To reduce latency, you can alter the number of slices one at a time. After migrating from the previous table count to the newly completed one, switch to the new count.

After all the data from the previous slicing count is moved to the new slicing count, make sure to remove the previous count from PREVIOUS_UM_CONTEXT_TABLE_SLICING_COUNT.

If PREVIOUS_UM_CONTEXT_TABLE_SLICING_COUNT does not contain 0, and ENABLE TABLE SLICING parameter is set to true, the data cannot be searched on the primary table.

Slicing in GxSession database for PCRF Core service
While upgrading from an older version of Policy to a later version, you can configure the Database Slicing feature for GxSession database as follows:

1. Enable the table slicing for GxSession database using GX_SESSION_TABLE_SLICING_ENABLED Helm parameter. This enables the database slicing on GxSession database. By default, the value of this key is set to false. Once the database slicing is enabled, the database operations are processed on the sliced tables.
2. When DISTRIBUTE_GX_TRAFFIC_USING_TABLE_SLICING advanced settings key is enabled, the Audit Service audits the GxSession sliced tables as well.
3. Configure the number of slices using GX_SESSION_TABLE_SLICING_COUNT Helm parameter. By default, the value of this parameter is set to 1.
4. In CNC Console, under **PCRF Core Settings** page in **Service Configurations**, set the value of DISTRIBUTE_GX_TRAFFIC_USING_TABLE_SLICING advanced settings key to true. By default, the value of this key is set to false. When this key is enabled and if the new slices for Gx are created, traffic for GxSessions are distributed across all the new tables. PCRF Core service will register the new tables with Audit service and also distributes the new sessions to the new slices.

> **Note**
>
> DISTRIBUTE_GX_TRAFFIC_USING_TABLE_SLICING advanced setting key must be enabled after upgrade or install with table slicing processes are done.

Identifying the slice for database operation:

- When PCRF Core service receives a CCR-I message, the primary key is hashed to find the slice. To save the GxSession in the database, use the primary key in the slicing function. The slicing function provides the slice number where the association can be saved.

- When PCRF Core service receives a CCR-U message, the slice number is obtained using the slicing function. Note that the Session-Id must never be changed. To update the database, the Session-Id is used with the slicing function to obtain the slice number where the association is saved.
- When PCRF Core service receives a CCR-T message, the slice number is calculated using the slicing function. The Session-Id is passed to the slicing function to obtain the slice number where the association to be deleted can be found.

Note

PCRF Core does not support mismatch of versions in multi-site configurations. Due to the limitations from changing the table name from GxSession to GxSession_n, multi-site configurations with different versions of PCRF installed is not supported.

Note

- You can start with a fixed number of slices that is estimated to cater to the replication needs of the near future (for example, next 5 years).
- All table slices may not have their own replication channels to start with. For example, to enable 75K TPS, only 2 replications channels can be configured, each replicating 4 slices. This configuration can change depending on the increase in TPS and available bandwidth across the replicating sites.
- Records are distributed across all the slices. This helps in reconfiguring the slice to replication channel mapping. No migration is required as long as the table slice count does not change.
- The hashing function selected is `Consistent Hash`.
- The hashed slice number is provided to the consumer service accessing the database as part of the resource-id to avoid the requirement of migration if and when the table slices need to be increased.
- For secondary key lookup, a parallel search is performed across all the table slices.

> **Note**
>
> Currently, Policy does not support mismatch of versions in multi-site configurations.
>
> Also, when Policy is upgraded from one version to the same version, the upgrade procedure will not trigger creation of database slices. Creation of slices must be enabled and configured using Helm. For more information, see *Preupgrade Tasks* section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.
>
> The following are some of the scenarios of database slicing during a Policy software upgrade in a geo-redundant setup:
>
> - If all the sites are using an older version of Policy and the first site is upgraded to a release that supports table slicing, but slicing is disabled. In this case, there is no need to change anything in the slicing configuration. To activate the distribution on the new slices DISTRIBUTE_GX_TRAFFIC_USING_TABLE_SLICING Advanced Setting key must be enabled.
> - If all the sites are upgraded to a newer version of Policy that supports table slicing, but slicing is enabled only on site1, only on the site 1, the sessions will be distributed across the slices.
> - When a new site is added to existing cluster where slicing is disabled, whereas slicing is enabled in other sites, only the cluster with table slicing enabled will have the distribution of sessions across all the slices. The other clusters will have only one table for the database.
> - When a new site is added to existing cluster with slicing enabled, whereas slicing is still disabled on existing sites, only on the new site the sessions are distributed across all the slices.

Observability

Metrics

Slicing SMPolicyAssociation for SM service

occnp_db_overall_processing_time. getBySecondaryKeyV4 metric is used to calculate total response time for secondary key serial search when table slicing on SMPolicyAssociation database is enabled.

Slicing UMContext for Usage Monitoring service

usage_mon_context_found metric is used to count the session lookup on the main table or the sliced tables for old sessions of the subscribers when table slicing is enabled.

4.108 Support for Interworking Between Evolved Packet Core (EPC) and CNC

This section describes how Converged Policy supports interworking between Evolved Packet Core (EPC) and 5G Cloud Native Core (CNC).

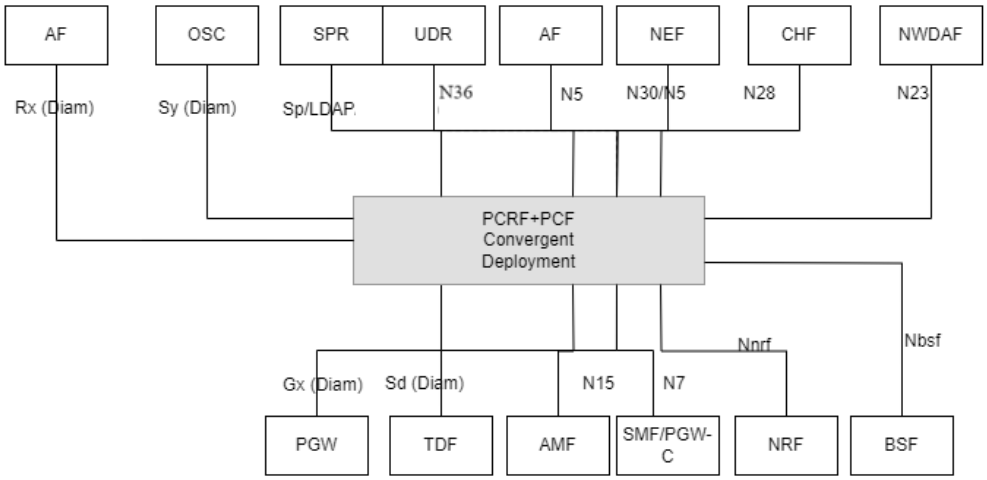
EPC provides converged voice and data services on a 4G LTE network to enable advanced services, such as VoIP.

EPC comprises:

- Serving Gateway (SGW)
- PDN Gateway (PGW)
- Mobility Management Entity (MME)
- Home Subscriber Server (HSS)

Converged Policy deployment to support both EPC/5G Core network element

Figure 4-178 Converged Policy Architecture to Support EPC and 5GC



3GPP recommends the following deployment Options with both LTE,NR and EPC,5GC co-exist. The Interfaces on which the UE attach lands for each of these options is shown.

Table 4-71 Deployment Options

3GPP Deployment Options	CnPolicy (PCRF+PCF)
Option 1: Standalone LTE under EPC	UE attaches to CnPolicy over Diameter Gx Interface.
Option 2: Standalone NR under 5GC	UE attaches to CnPolicy over Nsmf/N7 Interface.
Option 3: NonStandalone LTE and NR under EPC	UE attaches to CnPolicy over Diameter Gx Interface.
Option 4: NonStandalone NR and LTE under 5GC	UE attaches to CnPolicy over Diameter Gx Interface.
Option 7: Non Standalone LTE and NR under 5GC	UE attaches to CnPolicy over Nsmf/N7 Interface.

Typical Migration strategy from 4G LTE to 5GC SA is as follows:

4G LTE -> NSA Option 3 -> NSA Option 7: NSA Option 4 -> SA Option 2

5

Integrating Policy with Different Network Functions

You can integrate Oracle Communications Cloud Native Core, Converged Policy with different Network Functions (NFs), such as Oracle Communications Cloud Native Core, Network Repository Function (NRF), Oracle Communications Cloud Native Core, Unified Data Repository (UDR), and Charging Function (CHF).

NRF Integration

NRF Management (Client) service enables policy solution to integrate with NRF server for service registration, discovery, and service status/ load related information.

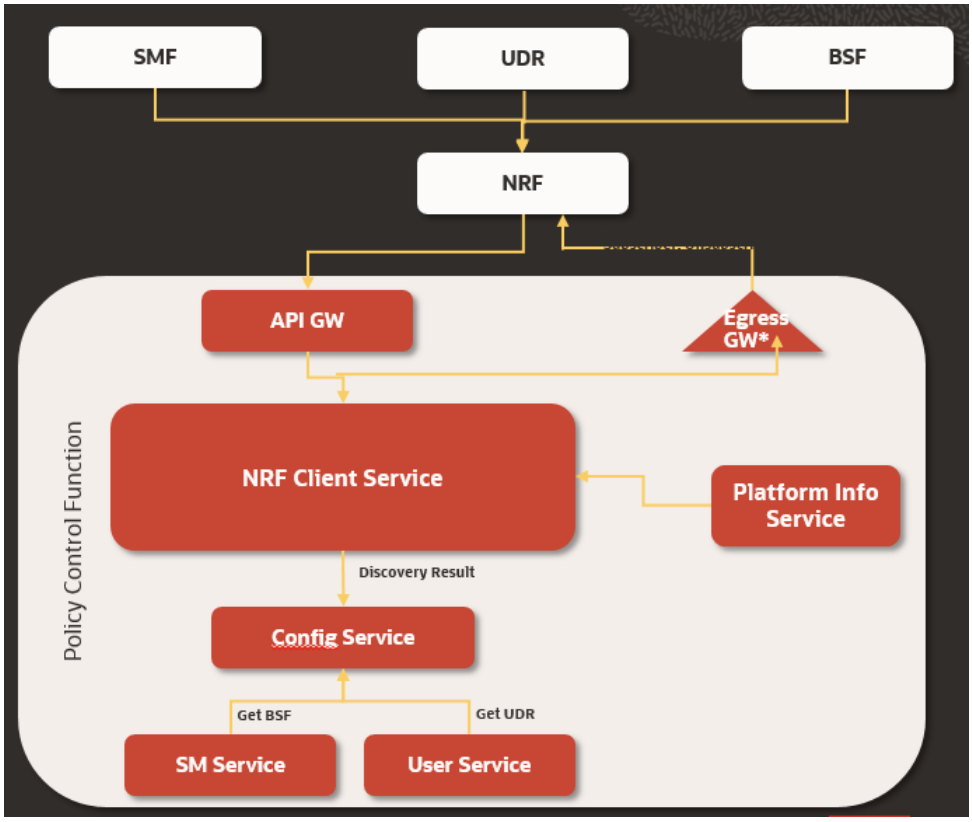
Management Service support includes the following functions:

- Register Service
- Deregister Service
- PCF heartbeat to NRF that includes load, priority, and capacity information
- Knowledge to NRF of scaling change
- Subscribe or Unsubscribe

Discovery Service

- Used to discover UDR, BSF, and CHF services
- Compliant with 29.510

Figure 5-1 NRF Integration



Editing Kubernetes Configuration Map

A Kubernetes Configuration Map is provided to save the NRF address and the NF Profile information. You can edit the Kubernetes Configuration Map to register Policy Control Function (PCF) with the NRF.

To edit the Kubernetes Configuration Map:

Open a console to the master node of the Kubernetes deployment and edit the config map named "*pcf-name*-application-config" where *pcf-name* is the HELM chart release name used at the time of installation, see *Oracle Communications Cloud Native Core Policy Installation, Upgrade and Fault Recovery Guide*.

1. Get a list of all the config maps in the PCF deployment namespace by running the following command:

   ```
   kubectl get cm -n pcf-namespace
   ```

   where, *pcf-namespace* is the PCF deployment namespace used by the helm command.
2. Edit the application configuration map by running the following command:

   ```
   kubectl edit cm pcf-name-application-config -n pcf-namespace
   ```

   where, *pcf-name* is the release name used by the helm command.
   A standard unix vi editor is opened with the config map contents pre-filled. Use vi commands to edit the application configuration map.

3. Verify the NRF address (fqdn/IP) and the port number. NRF address is contained in the custom value yaml file. See the `configmapApplicationConfig` attribute in the custom yaml file.
4. Check and add necessary NFs to "nrfClientSubscribeTypes". These NFs are discovered and subscribed by PCF at the startup time. Leave this field empty if this onetime discovery and subscription for NFs is not required.
5. Check and edit the PCF Profile to be registered with the NRF. For example, if required enter the IP details of the PCF Services.
6. Save and exit the editor.

UDR and CHF Integration

Policy solution supports integration with external policy data sources using user service encapsulates all the DB integration complexity from other micro-services. The feature helps the dynamic discovery of UDR and CHF from NRF and Nudr/Nchf interfaces.

Support for CHF to access counter information

- This is an evolution of the Sy interface, where the PCF consumes the Nchf_SpendingLimitControl service provided by the CHF.
- The service enables the PCF to retrieve policy counter status information per UE from the CHF by subscribing to spending limit reporting, such as notifications of policy counter status changes.
  - Dynamic discovery of CHF from NRF
  - Support for policy counter retrieval and subscription for changes and notification handling
  - Compliant with 29.594 v15.2.0

Support for UDR
This is an evolution of the 4G UDR/SPR where the PCF is able to retrieve, update, subscribe, and get notified to changes for:

- Session Management Policy Data
- Access And Mobility Policy Data
- UE policy data
- Usage Monitoring Data
- Policy Data Subscriptions
- Individual Policy Data Subscription
- Compliant with 29.519 V15.2.0

6
Configuring Policy

This section provides the information for configuring Oracle Communications Cloud Native Core, Converged Policy for various services.

Policy offers the following interfaces to configure the solution:

- A web-browser based Graphical User Interface (GUI)
  For more information on configurations using GUI, see Configuring Policy Using CNC Console.
- A REST API based Machine-to-Machine interface
  For more information about Policy REST APIs, see *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.
- Custom Value files to customize the Policy deployment.
  The Policy deployment can be customized by overriding the default values of various configurable parameters in the custom value files. For more information about downloading and customizing the custom value files, see *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.
- Kubernetes Configuration Maps. This configuration map is used to register PCF with NRF.
  For more information, see Integrating Policy with Different Network Functions.

7
Configuring Policy Using CNC Console

This chapter describes how to configure different global and service parameters in Oracle Communications Cloud Native Core, Converged Policy using Oracle Communications Cloud Native Core, Cloud Native Configurations Console (CNC Console).

Oracle Communications Cloud Native Core, Cloud Native Configuration Console Interface

This section provides an overview of the CNC Console, which includes an interface to help in creating global and service parameters in Policy.

You can use Policy integration with CNC Console only after logging successfully in to the CNC Console application. To log in to the CNC Console, make the following updates to the hosts file available at the `C:\Windows\System32\drivers\etc` location.

1. In Windows system, open the **hosts** file in a notepad as an Administrator and append the following set of lines at the end:

   ```
   <IP Address> cncc-iam-ingress-gateway.cncc.svc.cluster.local
   ```

   ```
   <IP Address> cncc-core-ingress-gateway.cncc.svc.cluster.local
   ```

   where:

   <IP Address> is the host address of the deployment cluster. It depends on the deployment cluster.

   Example:

   ```
   10.75.225.189 cncc-iam-ingress-gateway.cncc.svc.cluster.local
   ```

   ```
   10.75.225.189 cncc-core-ingress-gateway.cncc.svc.cluster.local
   ```

   > **Note**
   >
   > The IP Address can change when deployment cluster changes.

2. Save and close the hosts file.

> **Note**
>
> Before logging into CNC Console, create a CNC user and password. Using these user details, you can log in to the CNC Console application. For more information about creating a CNC Console user and password, see *Oracle Communications Cloud Native Core, Cloud Native Configuration Console Installation, Upgrade, and Fault Recovery Guide*.

To log in to CNC Console:

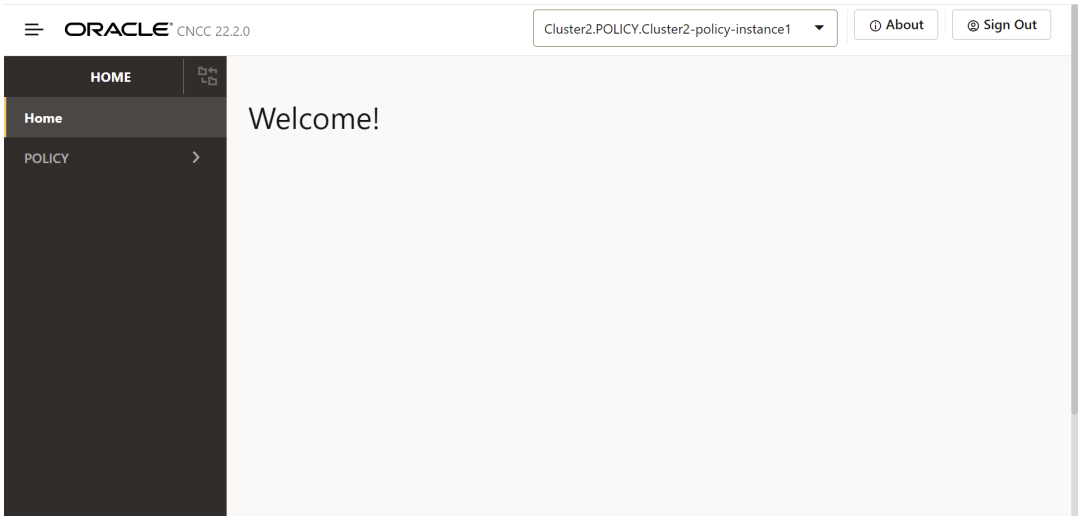
1. Open a web browser and enter the URL: http://cncc-core-ingress-gateway.cncc.svc.cluster.local:*port number*/ and press Enter.

> **Note**
>
> *port number* is cncc-iam-ingress-port number.

The login page opens.

2. Enter the **Username** and **Password**.
3. Click **Log In**.
4. On the Welcome page, select the required NF instance from the **Please Select Instance** drop-down field.
   This opens the CNC Console home page for the selected NF instance:

Figure 7-1 CNC Console for Policy



5. To use Policy services integrated with CNC Console, click **Policy** in the left navigation pane.

7.1 General Configurations

This section describes how to customize the general settings in a Policy deployment using the **General Configuration** pages. The configurations include general settings, such as log levels settings, enabling or disabling metrics, subscriber activity logging settings, and so on.

7.1.1 General Settings

The **General Settings** page displays the General Settings related to the Policy setup. The page allows you to edit the configurations.

To edit the General Settings:

1. From the navigation menu, under **Policy**, navigate to **General Configurations**, and select **General Settings**.
   This opens the **General Settings** page. The page displays the existing configurations.
2. Click Edit .
   This opens the **Edit General Settings** page.

3. Enter the following information:

Table 7-1 Edit General Settings

Field Name	Description
apiGatewayHost	Specifies the name for the API gateway host .
apiGatewayPort	Specifies the port number of the API gateway .
Enable Tracing	Specifies whether to enable/disable tracing. The default value is `true`.
Enable Metrics	Specifies whether to enable/disable system metrics. The default value is `true`.
Enable TLS	Specifies whether to enable/disable TLS. The default value is `false`.
Enable Subscriber Activity Logging	Specifies whether to enable/disable subscriber activity logging. The default value is `false`.
Enable Policy Event Record	Specifies whether to enable/disable Policy Event Record (PER) feature. The default value is `false`.
Policy Event Record Host	Specifies the valid URL of PER host to receive the PER record. The format of the url is: http://*per-host*:*per-port*, where *per-host* specifies the PER host and *per-port* specifies the PER port. For example, `http://localhost:8101/v1/echo`
Enable SBI Correlation	This specifies whether to enable/disable correlation-info header in PCF. The dafault value is false.

4. Click **Save**.
The page saves the General Settings.

7.1.2 Logging Configurations

This section describes how to customize the log level and subscriber logging activity settings in Policy using the **Logging Configurations** pages.

7.1.2.1 Logging Level

This procedure describes how to configure log level for different Policy services through CNC Console.

> **Note**
>
> Default log level for each service is Warn.

The **Logging Level** page displays the log level configured for different Policy services. The page allows you to edit the log level configurations.

To configure the log level:

1. From the navigation menu, under **Policy**, navigate to **Logging Configuration**, and select **Logging Level**.
   This opens the **Logging Level Configuration** page. You can add, edit, or delete the log level and package log level for each service type from this page.
2. Click 

   .
   This opens the **Edit Log Level** page.
3. From the **Service Type** drop-down list, select the service for which you need to view, edit, or delete the logs.
4. From the **Application Log Level** drop-down list, select the root log level of the application for the selected service type. Possible values are:
   - **TRACE**
   - **DEBUG**
   - **INFO**
   - **WARN**
   - **ERROR**

   > **Note**
   >
   > The value for the **Application Log Level** field is the mandatory value, and the package log level is the optional value.

5. Expand the **Package Log Level** group to enter the package log level information:

   > **Note**
   >
   > This section is only applicable when Oracle Engineering is trying to isolate an issue and requests one or more package names be added and logs collected after the reproduction of an issue.

   a. Click 

      .
      The **Add Package log Level** dialog box opens.
   b. Enter the value in the **Package** field.
      The value of **Package** field is dependent on the name of the package in each application. Before you set value of the **Package** field, you must know which package is available in that application.
   c. From the **Log Level** drop-down list, select the log level for the package. Possible values are:
      - **TRACE**
      - **DEBUG**
      - **INFO**
      - **WARN**
      - **ERROR**
   d. Click **Save**.
      The Package log level information for the selected service is saved.

> **Note**
>
> Use the Edit or Delete icons available in the next column to update or delete the package log level information.

6. Click **Save**.
   The page saves the log level information for the selected service type.

7.1.2.2 Subscriber Activity Logging

Subscriber Activity Logging allows you to define a list of the subscribers (identifier) that you are may require to troubleshoot the NFs and trace all the logs related to the subscribers separately to view. This functionality can be used to troubleshoot problematic subscribers without enabling logs or traces that can impact all subscribers. You can capture and monitor subscriber logs for UDR or CHF notifications, and associated call flow in Session Management (SM), Access and Mobility (AM), User Equipment (UE), PCRF Core, and Egress Gateway.

To enable the subscriber activity logging functionality, set value of the **Enable Subscriber Activity Logging** parameter to **true** on the **General Configurations** page. By default, this functionality remains disabled. For more information about enabling the functionality, see General Settings.

This procedure provides information about how to configure and manage subscriber logging.

The **Subscriber Activity Logging** page allows you to create new and manage existing subscribers. The page displays the list of defined subscribers and provides the options to import, export, or add lists.

You can configure the list of subscribers using the **Subscriber Activity Logging** page.

To configure a list of subscribers for logging:

To configure Subscriber Activity Logging:

1. From the navigation menu under **Policy**, navigate to **General Configurations**, click **Logging Configurations**, and then select **Subscriber Activity Logging**.
   This opens the **Subscriber Activity Logging** page. The page lists the existing configurations. You can add or import new subscriber activity logging configurations using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Subscriber Activity Logging** page.
3. On the **Create Subscriber Activity Logging** page, enter the following information:

Table 7-2 Create Subscriber Activity Logging

Field Name	Description
Identifier Type	Select the subscriber identifier type. Supported subscriber identifier type are: • GPSI • SUPI • IPV4 • IPV6 **Note** Subscriber Activity is supported for 64/128 prefix. **Note**: AM and UE services do not support IPV4 or IPV6 identifiers.
Identifier Value	The identifier value for the selected identifier type.
Enable	Use this switch to enable or disable the subscriber logging functionality for the selected subscriber.

4. Click **Save**.
   The configuration gets listed on the **Subscriber Activity Logging** page. The page defines the Subscriber Activity Logging configuration in the Policy database and it is available to be used in a Policy.

> **Note**
>
> Use or available under the **Actions** column to update or delete the configuration.

Importing Subscriber Activity Logging

To import Subscriber Activity Logging configuration:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

Subscriber Identifiers

SUPI

In the 5G system, a globally unique Subscription Permanent Identifier (SUPI), known as IMSI (International Mobile Subscriber Identity) till 4G, is assigned for each subscription. The SUPIs are assigned in such a manner that it helps in identifying subscriptions and is independent of the user equipment.

For 4G systems, the value of IMSI is structured as:

imsi: <value>

For 5G systems, the value of SUPI is structured as:

supi: imsi-<value>

GPSI

General Public Subscription Identifier (GPSI), known as MSISDN (Mobile Station International Subscriber Directory Number) till 4G, is a 3GPP defined subscriber public identifier that can be used both inside and outside of the 3GPP system. The association between GPSI and its related SUPI are stored in the subscription data in a 5G system.

For 4G systems, the value of MSISDN is structured as:

msisdn/e164:<value>

For 5G systems, the value of GPSI is structured as:

gpsi: msisdn-<value>

Limiting size of the Subscriber Activity Logging Mapping Table

Users can specify the number of sessions in the subscriber activity mapping table. By default, the number of sessions per subscriber is defined as 20 in the mapping table.

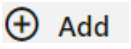
However, the users can modify the number of sessions by changing the value on the `SUBS_ACT_MAPPINGTABLE_ENTRY_SIZE` through CM service, Ingress Gateway deployment, or Diameter Gateway.

Limiting size of the mapping table helps in maintaining the network latency and size of the mapping table.

7.1.3 SBI Ingress Error Code Profiles Collection

This procedure provides information about how to use the SBI Ingress Error Code Profiles Collection page to create and manage SBI Ingress error code profiles collection in General Configurations.

To configure Error Code profiles collection, perform the following steps:

1. From the navigation menu, under **Policy**, click **General Configurations**, and then select **SBI Ingress Error Code Profiles Collection**.
   This opens the SBI Ingress Error Code Profiles Collection page.
2. Click Edit.
   This opens the Edit SBI Ingress Error Code Profiles Collection page.
3. Click 

   .
   This opens the Add SBI Ingress Error Code Profiles Collection page.
4. Enter values for the available input fields as described in the following table:

Table 7-3 Error Code Profiles Configurations

Field Name	Description
Name	Specifies a unique name to identify the error profile.
Error Code	Specifies the HTTP Code that is populated in the error response when a message request is rejected due to overload control.

Table 7-3 (Cont.) Error Code Profiles Configurations

Field Name	Description
Error Cause	Specifies the error cause that is populated in the error response when a message request is rejected due to overload control.
Error Title	Specifies the error title that is populated in the error response when a message request is rejected due to overload control.
Error Description	Specifies the error description that is populated in the error response when a message request is rejected due to overload control.

5. Click **Save** to save the error code profile.
   To discard the changes, click **Cancel**

   The value gets listed on the SBI Ingress Error Code Profiles Collection page. Use or available under the **Actions** column to update or delete the profile.

7.2 Error Handling

This section describes how to manage and view the error configurations in Policy, using the **Error Handling** Configurations page.

7.2.1 Error Configurations

The error handling framework allows the users to configure an error state and an action for it. The action contains two parts, an error rule and an error context. On the Console UI, the operator configures the error state specific to the PCF services and the list of actions for it.

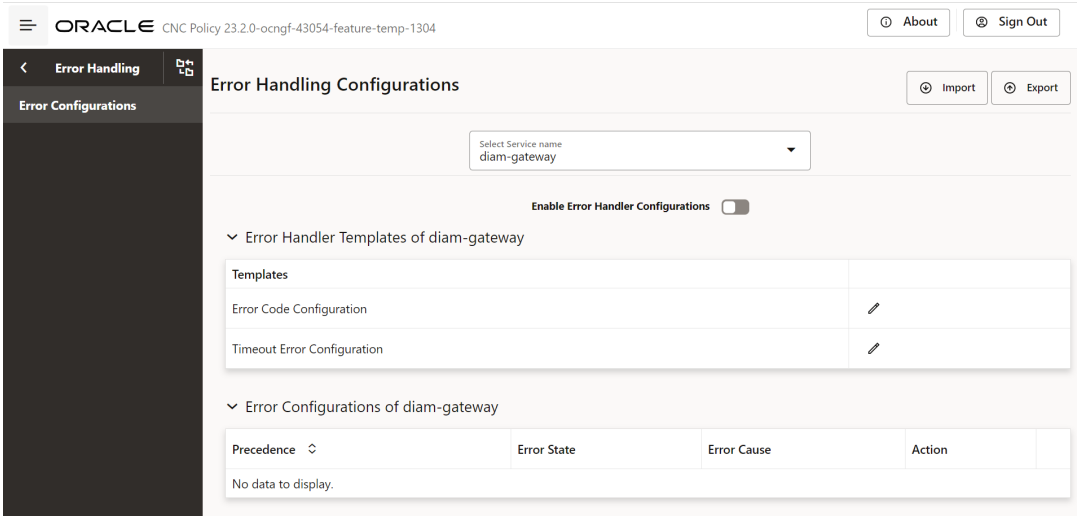
The **Error Configurations** page displays the error configurations related to different Policy services.

To add error handler template for a Policy service:

Error Configurations for Diameter Gateway Service

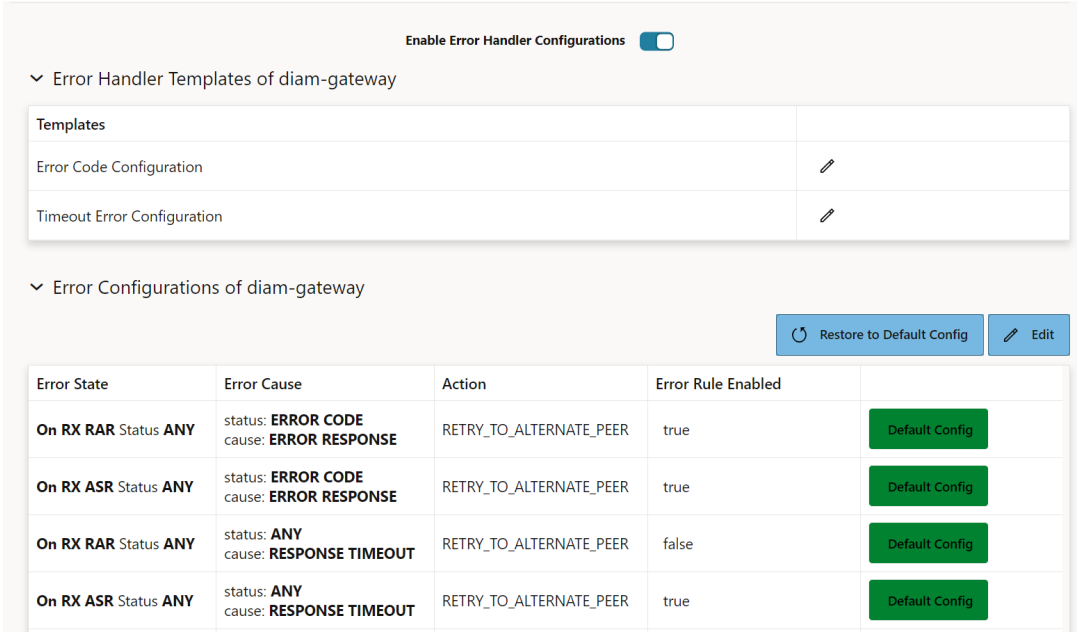
1. From the navigation menu, under **Policy**, click **Error Handling**, and select **Error Configurations** page. This opens the **Error Handling Configurations** page.
2. From the `Select Service Name` drop-down list select the value `diam-gateway`. The page allows you to add and edit configurations for diameter message retry for Rx RAA or Rx ASR. On the page **Error Handler Templates of diam-gateway** and **Error Configurations of diam-gateway** subsections are displayed.
3. Enable the error handler configurations using the **Enable Error Handler Configurations** toggle button.
4. The **Error Handler Templates of diam-gateway** provides two options:
   - **Error Code Configuration**
   - **Timeout Error Configuration**

Figure 7-2 Error Handling Configuration UI



5. The **Error Configurations of diam-gateway** provides default error handling configurations to retry on all error codes (except diameter result code 2xxx) and timeout for Rx RAA and Rx ASA failed diameter messages.

Figure 7-3 Default Configurations



6. To configure the **Error Code Configuration** in **Error Handler Templates of diam-gateway** , Click Edit .
   This opens the **Error Handler Template** editing page.
7. Enter values for the available input fields. The following table describes the fields:

Table 7-4 Create Error Code Configuration - Edit

Field Name	Description
On Rx	Specifies the list of diameter interfaces. The values are: • RAR • ASR Default value: RAR
Status	Specifies the error status to be provided by the user.
Error Cause Configure	
Error Cause Field	Species to search for which error causing filed in the diameter answer message. Default value: ALL
Match Operator	Species the match operator to search error section in the diameter answer message. Default value: ANY
Message	Specifies the error message to search in the error section of diameter answer message. Default value: ANY
Status	Specifies the error status code to search in the error section in the diameter answer message. Default value: ANY
Cause	Specifies the field that matches the cause during error ends with 'not found'. User can choose from the following options: • ANY • RESPONSE_TIMEOUT Default value: ANY
Action	
Action	The action to be performed in the event of failed diameter message on Rx interface. User can choose from the following options: • RETRY TO ALTERNATE PEER • ONE RETRY TO ALTERNATE PEER
Error Originator	The peer from which the error origination occurs. User can choose from the following options: • ANY • INTERMEDIATE_PEER • DESTINATION PEER

Figure 7-4 Edit Error Code Configuration:

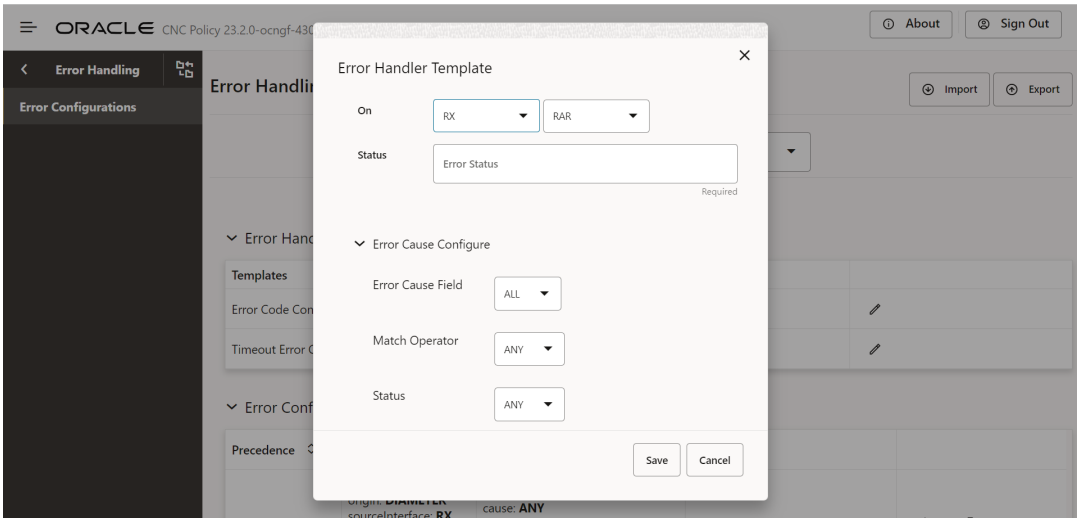
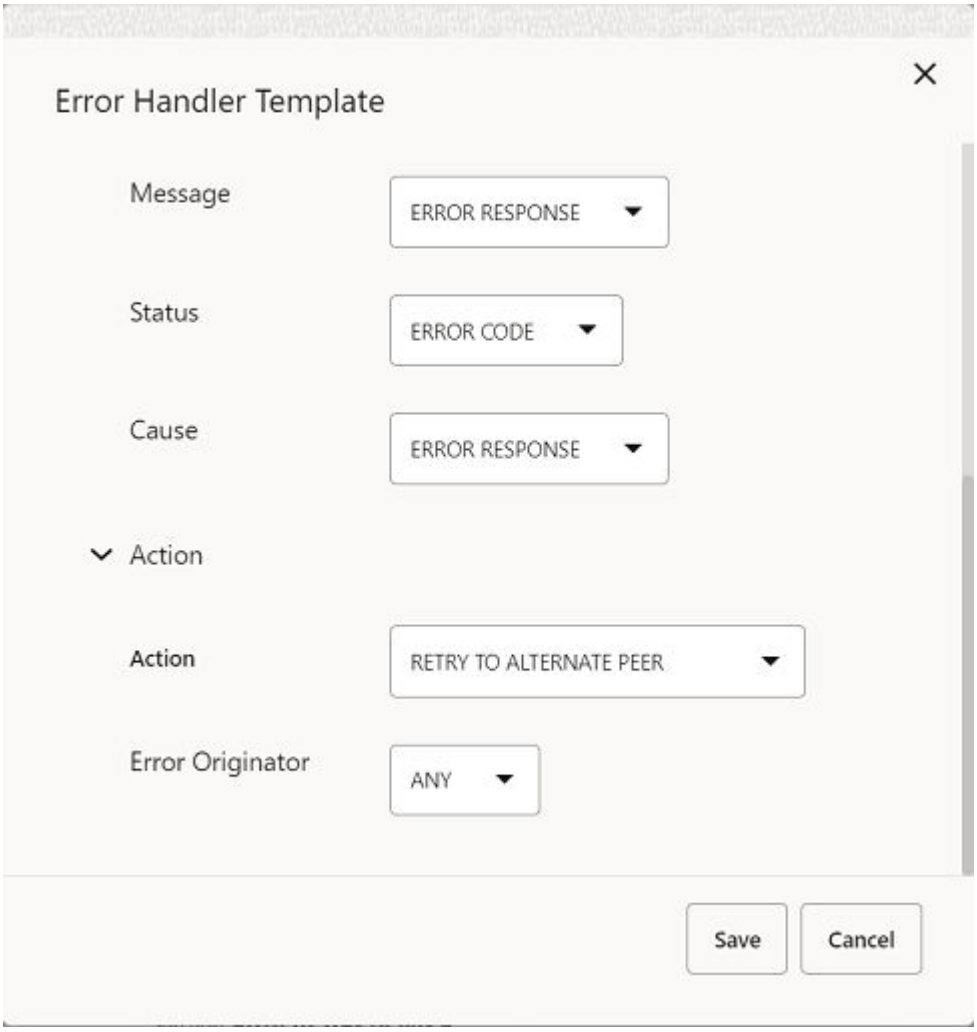


Figure 7-5 Edit Error Code Configuration continuation..



8. Click **Save** to save the changes or Click **Cancel** to discard the changes.

9. To configure **Timeout Error Configuration** in **Error Handler Templates of diam-gateway** , Click 

.
This opens the **Error Handler Template** editing page
10. Enter values for the available input fields. The following table describes the fields:

Table 7-5 Create Timeout Error Configuration

Field	Description
On RX	Specifies the list of diameter interfaces. Default value: RAR
Status	Specifies the error status to be provided by the user. Default value: ANY
Error Cause Configure	
Error Cause Field	Species to search for which error causing filed in the diameter answer message. Default value: MESSAGE
Match Operator	Species the match operator to search error section in the diameter answer message. Default value: EQUALS
Message	Specifies the error message to search in the error section of diameter answer message. Default value: TIMEOUT_EXCEPTION
Status	Specifies the error status code to search in the error section in the diameter answer message. Default value: ANY
Cause	Specifies the field that matches the cause during error ends with 'not found'. Default value: ANY
Instance	Specifies the field that matches the instance during error contains the term 'Illegal'. Default value: ANY
resource	Specifies the resource to search in the error section of diameter answer message. Default value: ANY
Action	
Action	The action to be performed in the event of response timeout on Rx interface. User can choose from the following options: • RETRY TO ALTERNATE PEER • ONE RETRY TO ALTERNATE PEER
Error Originator	The peer from which the error origination occurs. User can choose from the following options: • ANY • INTERMEDIATE_PEER • DESTINATION PEER

Figure 7-6 Edit Timeout Error Configuration:

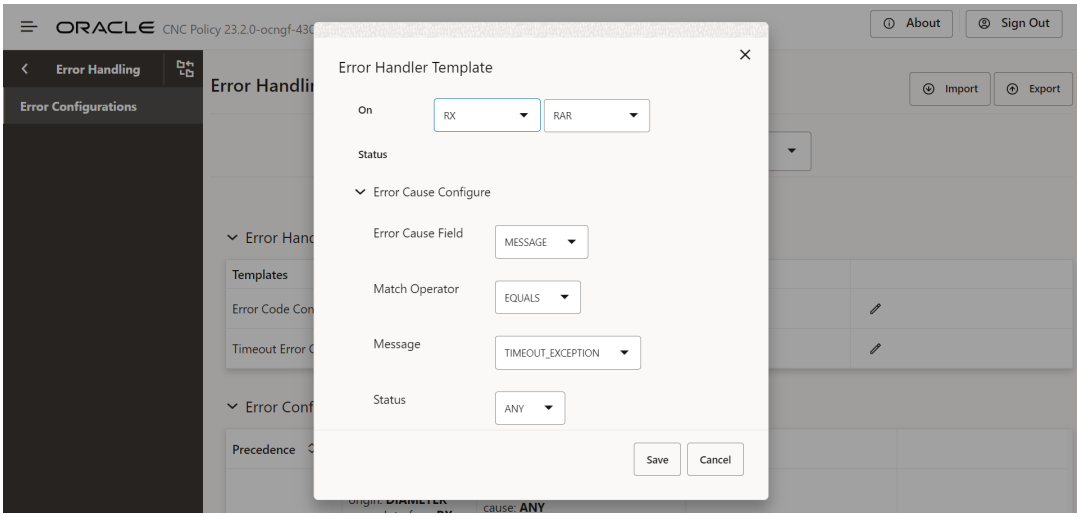
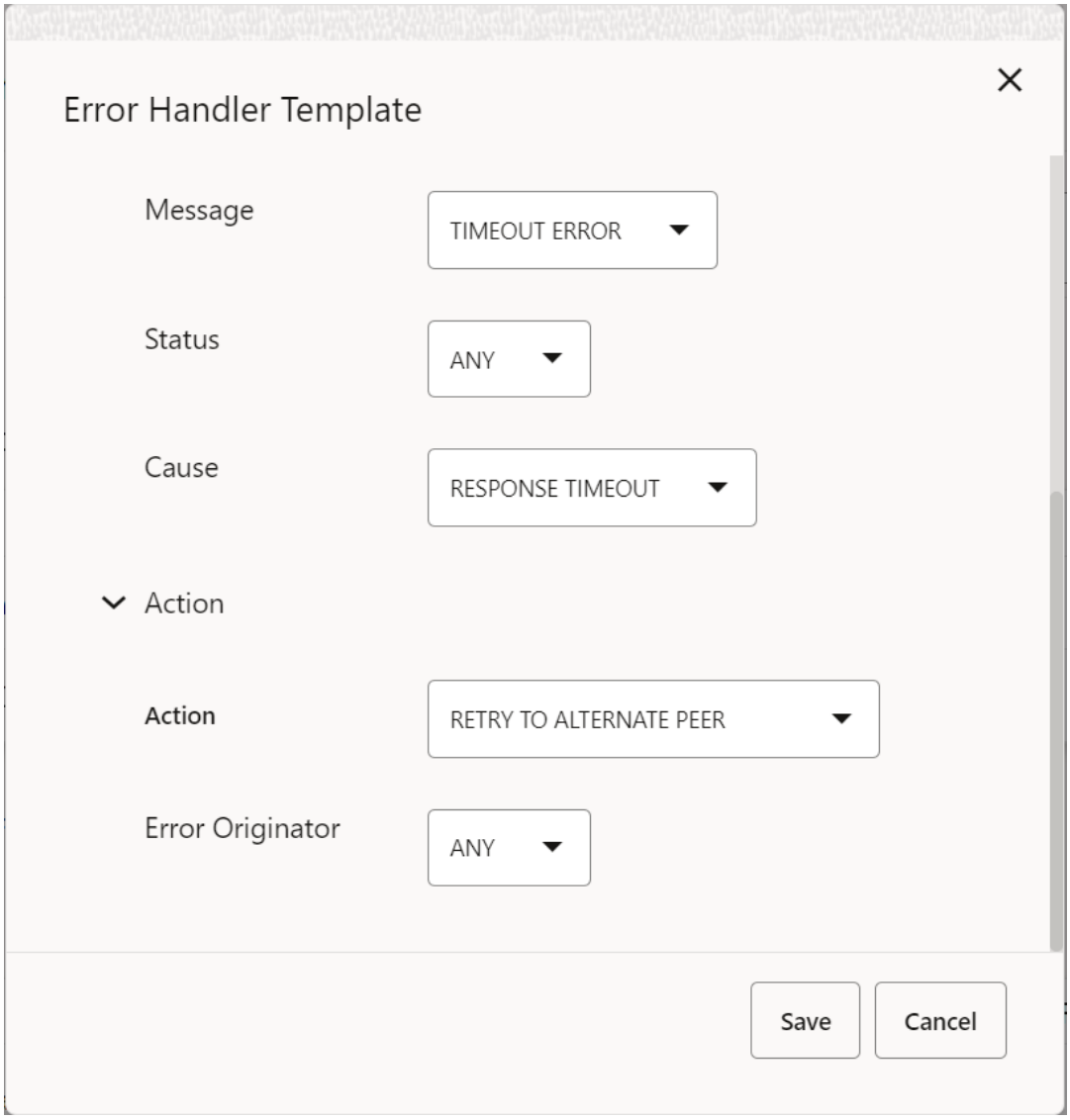


Figure 7-7 Edit Timeout Error Configuration Continuation:



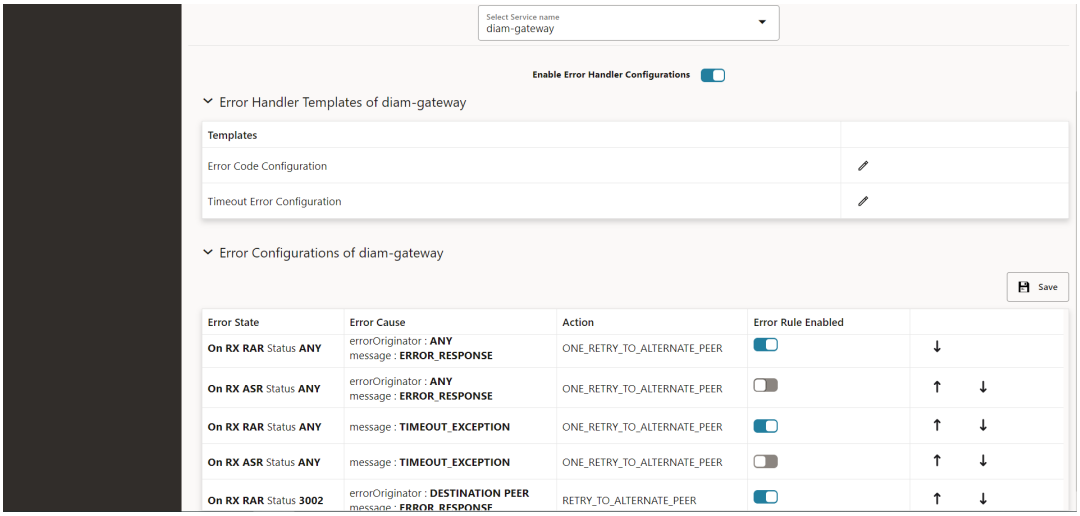
11. Click **Save** to save the changes or Click **Cancel** to discard the changes.
Using the edit error handling configuration option, you can enable or disable the configurations.

The priority for each error handling configuration can be set using the

and

arrow buttons.

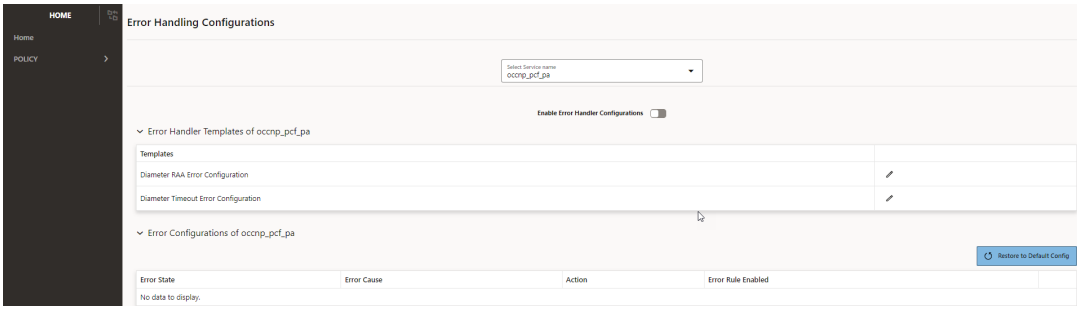
Figure 7-8 Error Handling User Saved Configuration with Priority:

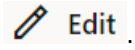


Error Configurations for PA Service

1. From the navigation menu, under **Policy**, click **Error Handling** and select **Error Configurations** page.
   This opens the **Error Handling Configurations** page.
2. From the `Select Service Name` drop-down list, select the value `occnp_pcf_pa`.
3. Enable the error handler configurations using the **Enable Error Handler Configurations** toggle button.
4. The **Error Handler Templates of occnp_pcf_pa** provide two options:
   - Diameter RAA Error Configuration
   - Diameter Timeout Error Configuration

Figure 7-9 Error Handling Configuration



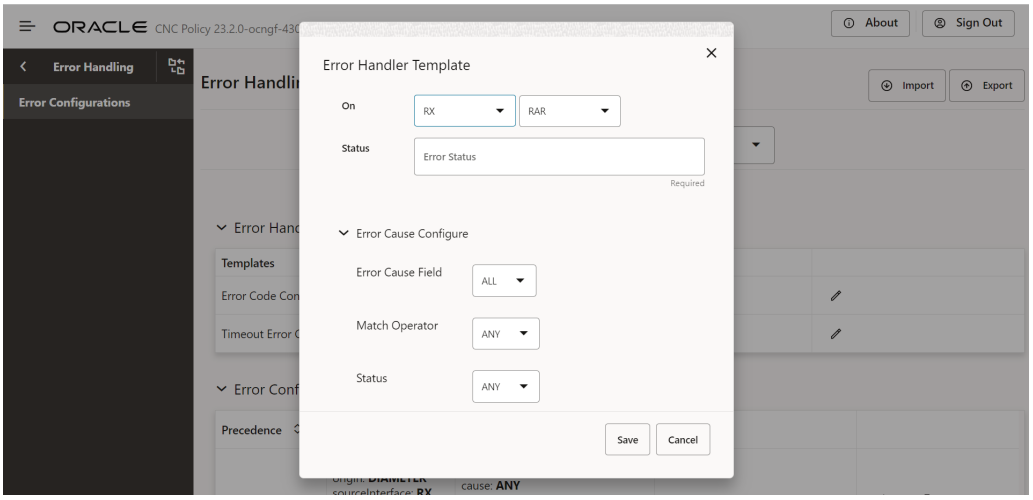
5. To configure the **Diameter RAA Error**, Click 

   .
   This opens the **Error Handler Template** editing page for Diameter RAA Error.
   a. Enter values for the available input fields. The following table describes the fields:

Table 7-6 Diameter RAA Error Configuration

Field Name	Description
On Rx	Specifies the list of diameter interfaces. **Default Value**: RAR
Status	Specifies the diameter error result code to be provided by the user.
Error Cause Configure	
Error Cause Field	Specifies the error causing filed in the diameter answer message. **Default Value**: Error Response Originator
Match Operator	Specifies the match operator to search error section in the diameter answer message. **Default Value**: EQUALS
Error Response Originator	Specifies the originator of the error response. **Default Value**: ANY
Action	
Action	Either terminate the ongoing transaction related to RAA or cleanup the Rx/N5 session. **Default Value**: Terminate Transaction

Figure 7-10 Diameter RAA Error Configuration:



b. Click **Save** to save the changes or Click **Cancel** to discard the changes.

6. To configure **Diameter Timeout Error Configuration**, Click Edit.
This opens the **Error Handler Template** page for Diameter Timeout Error Configuration.

a. Enter values for the available input fields. The following table describes the fields:

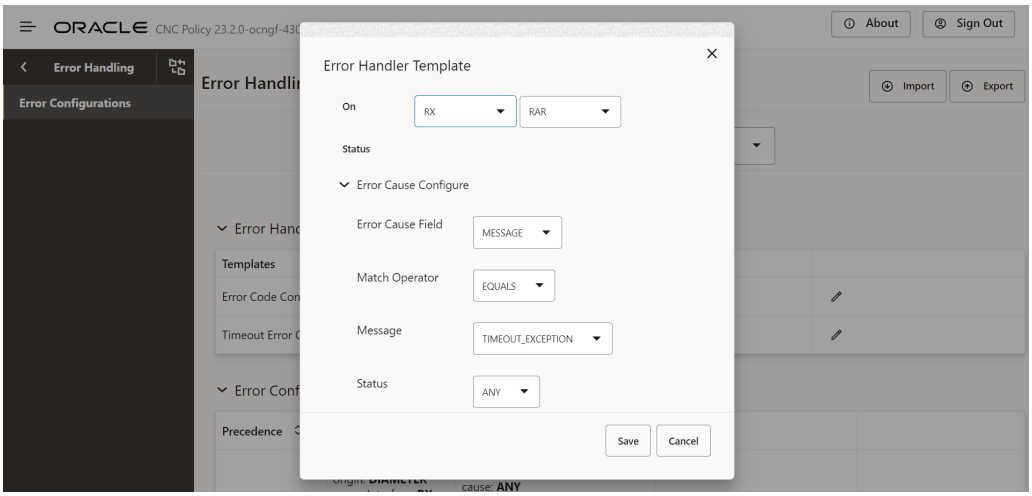
Table 7-7 Diameter Timeout Error Configuration

Field	Description
On RX	Specifies the list of diameter interfaces. **Default Value**: RAR

Table 7-7 (Cont.) Diameter Timeout Error Configuration

Field	Description
Status	Specifies the error status to be provided by the user. **Default Value**: NA
Error Cause Configure	
Error Cause Field	Specifies the error causing filed in the diameter answer message. **Default Value**: MESSAGE
Match Operator	Specifies the match operator to search for error section in the diameter answer message. **Default value**: EQUALS
Message	Specifies the error message to search in the error section of diameter answer message. **Default value**: TIMEOUT_EXCEPTION
Action	
Action	Either terminate the ongoing transaction related to RAA or cleanup the Rx/N5 session. **Default value**: Terminate Transaction

Figure 7-11 Diameter Timeout Error Configuration:



b. Click **Save** to save the changes or Click **Cancel** to discard the changes.

Error Configurations for AM, SM, UE, CHF Connector, UDR Connector, and Binding Services

1. From the navigation menu, under **Policy**, click **Error Handling**, and select **Error Configurations** page. This opens the **Error Handling Configurations** page.
2. From the **Select Service Name** drop-down list, select the required service.
3. Enable the error handler configurations using the **Enable Error Handler Configurations** toggle button.
4. The error handler template provides *Error Enhancement Configurations*.
5. To configure the *Error Enhancement Configurations* in Error Handler Templates of the required service, Click Edit. This opens the **Error Handler Template** editing page.

6. Enter values for the available input fields. The following table describes the fields:

Table 7-8 Error Handler Template

Field	Description
On	Specifies the Application Error. Default value: Application Error
Action	
Action	Specifies the action to be performed in the event of failed message. Default value: Reject with Enhanced Detail
Exclude from error message	Specifies exclusion of the provided components from detail error message. By default, "Error State and "Probelm Cause" are excluded.

7. Click **Save** to save the changes.

> **Note**
>
> Click **Cancel** to discard the changes.

7.3 Service Configurations

This section describes how to customize the Policy services according to the network requirements using the **Service Configuration** pages. The configurations include setting up end point addresses, setting up log levels, log level tracing, customizing and optimizing NF interactions, such as with UDR and so on.

> **Note**
>
> The advanced settings keys should have a unique value without any duplicates. For example, CONCURRENCY.BULWARK_ENABLED_FOR_CHF_NOTIFICATION settings for PDS should have a single key with value either true or false. There should not be two keys for CONCURRENCY.BULWARK_ENABLED_FOR_CHF_NOTIFICATION, each of them holding different values.

7.3.1 Common Data

This section includes the common data configurations for Policy services.

To access Common Data functionality from the CNC Console home page, expand **Policy**, navigate to **Service Configurations** and select **Common Data**.

On clicking **Common Data**, you can perform the following operations:

- Reattempts Profile
- Retry Profiles
- Timer Profiles
- Site Takeover

- NF Communication Profiles
- Attribute Forwarding Profiles

7.3.1.1 Reattempts Profile

This procedure provides information about how to create and manage the reattempts profiles for the binding registration.

The **Reattempts Profile** page allows you to create new and manage existing reattempt profiles. The page displays the list of defined reattempt profiles with the options to import, export, or add profiles.

To configure reattempt profile:

1. From the navigation menu under **Policy**, navigate to **Service Configurations**, click **Common Data**, and then select **Reattempts Profile**.
   This opens the **Reattempts Profile** page. The page lists the existing reattempt profiles. You can add or import new profiles using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Reattempts Profile** page.
3. On the **Create Reattempts Profile** page, enter the following information:

Table 7-9 Create Reattempts Profile

Field Name	Description
Name	The unique name for the reattempt profile. This name is used to refer to the reattempt profile in other service configuration screen, such as SM Service. The name can only contain the characters A–Z, a–z, 0–9, period (.), hyphen (-), and underline (_). The maximum length is 255 characters.
Interface Type	The interface for which a reattempt profile is being created. **Note:** Policy supports only the BSF Interface.
BSF Reattempts Settings	
Maximum Number of Attempts	The maximum number of allowed recreate attempts. The number of reattempts that must be run when PCF Binding request fails. **Range:** 0 to 50

Table 7-9 (Cont.) Create Reattempts Profile

Field Name	Description
Back-off timer configurations	The amount of time that represents each back-off timer. The **Back-off timer configurations** table includes the following parameters: • **Index:** Represents the number of back-off timer • **Back-Off Timer:** The timer value in seconds • **Max Random Number (Millisecond):** Any random number that is added to the back-off timer. **Range:** 0-5000 milliseconds **Note:** In case the number of configured attempts in the Back-off timer configurations table is less than the set value for **Maximum Number of Attempts**, the remaining attempts uses the back off timer and max random number of the last configured attempt. **Note:** Policy supports a maximum of 10 back-off timer configurations.
ThresHold Limit Level	The threshold limit for the Pending Operations. Once this value is reached, no pending operations are added to the table. For information about pending operation configurations, see PCF Session Management. **Default Value:** 1000

4. Click **Save**.
   The reattempt profile gets listed on the **Reattempt Profile** page. The page defines the reattempt profile in the Policy database and it is available to be used in a Policy.

> **Note**
>
> Use ✎ or 🗑 available under the **Actions** column to update or delete the reattempt profile.

Importing Reattempt Profiles

To import Reattempt Profile:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.3.1.2 Retry Profiles

This procedure provides information about how to create and manage retry profiles.

The **Retry Profile** page allows you to create new and manage existing retry profiles. The page displays the list of defined retry profiles with the options to import, export, or add profiles.

To configure retry profile:

1. From the navigation menu under **Policy**, navigate to **Service Configurations**, click **Common Data**, and then select **Retry Profiles**.
   This opens the **Retry Profiles** page. The page lists the existing retry profiles. You can add or import new retry profile using this page.

> **Note**
>
> Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Retry Profile** page.
3. On the **Create Retry Profile** page, enter the following information:

Table 7-10 Create Retry Profile

Field Name	Description
Name	The unique name for the retry profile. This name is used to refer to the retry profile in other service configuration screen, such as SM Service, User Connector and so on. The name can only contain the characters A–Z, a–z, 0–9, period (.), hyphen (-), and underline (_). The maximum length is 255 characters.
Retry on Internal Send Failure	Specifies whether to enable or disable retry for failed messages between core microservices of Policy. On enabling this switch, ensure that you configure the parameters under **Retry Settings** group. However, if the switch is disabled, the configurations in the Retry Settings group are not taken into consideration. When enabled, retries are attempted when PCF encounters internal send failures for egress messages. A failure to send egress messages internally indicates an exception in the core service, such as SM service, UDR or CHF connector, and so on. The exception can occur while connecting to the next microservice or a failure on the microservice itself. For example, egress message did not reach the egress gateway. This also includes the connection errors and timeouts on the core microservice. **Note**: Since the failure is internal, the external NF and SCP (if applicable) remain the same during retry attempt.
Enable Alternate Routing	Specifies whether to enable or disable alternate routing. On enabling this switch, ensure that you configure the parameters under the Alternate Routing Settings group. When enabled, alternate routing is attempted when the Policy messages are not delivered successfully to an external NF destination. It may happen because of an exception on the Egress Gateway, such as connection error, timeout, or a failure returned by the external NF or an intermediate router.

Table 7-10 (Cont.) Create Retry Profile

Field Name	Description
Server Header Support	Specifies whether to enable or disable server header. The value configured in this parameter is used when PCF acts as a consumer and receives server header in error response messages. Depending on the requirements, you can select any of the following values from the drop-down list: • Disabled (**default**) - Select this option to disable server header. On disabling server header, Policy as a consumer searches for alternate producer without considering the value of server header. • Single Instance - Select this option to accept only one value in the server header. When the value is configured as Single Instance and server header contains multiple values, Policy rejects the header. • Multi Instance - Select this option to accept multiple values in the server header.
Pattern To Ignore Server Header Value	Specifies the pattern to ignore server header values. **Note** This field appears only when *single Instance* or *Multi Instance* is selected fron the drop-down of the *Server Header Support* field.
Ignore Custom server header value	This field appears on selection of either "Single Instance" or "Multi Instance" value for the field **Server Header Support**. In this field you can specify the string pattern that can be ignored in the server header value. This is provided in the form of regular expression. The regular expression evaluates the server header content to ignore the strings that match the pattern specified the regular expression.
Retry Settings	Provide Retry Settings details:
Maximum Number Of Retries	Specifies the number of retries that PCF attempts on encountering failure while sending egress messages from Policy core microservices. **Note**: The value configured for this parameter does not include the initial attempt. For instance, if you set the value as 3, Policy performs the retry cycle three times after the initial attempt.
Alternate Routing Settings	Provide Alternate Routing Settings details:
Maximum Number Of Alternate Routing Attempts	Specifies the number of retries that PCF attempts on receiving error response from Egress Gateway. The number of retry attempts can be in the range of 1 to 10. **Note**: The value configured for this parameter does not include the initial attempt. For instance, if you set the value as 3, Policy performs the retry cycle three times after the initial attempt. If the retry attempts are exhausted, CNC Policy fails the transaction with an exception. In certain cases, the number of retry attempts can be greater than the available alternate destinations to be tried. If such a situation arises, Policy fails the transaction when no alternate destination is available even if retry attempts are still available.

Table 7-10 (Cont.) Create Retry Profile

Field Name	Description
Attempt Alternate Route for Following Error Codes	Allows to configure the HTTP error codes for which Policy reattempts the requests after the ARR request failures. **Note**: In Policy 22.1.0, Error Causes does not work for Notify Flows. Users are recommended to leave it blank. To add a value, click **Add** and enter values for the following fields: • **Error Code** - Specifies the HTTP error code. For example, 504 • **Error Causes** - Specifies the error cause. For example, GATEWAY_TIMEOUT, NF_TIMEOUT. You can configure up to 32 error codes with each error code supporting up to five causes.
Priority Pool	Specifies whether to enable or disable the load sharing between different NFs.
Use Alternate SCP for Alternate Routing	Specifies whether to enable or disable choosing alternate SCP to reach the alternate destination. If you set the value for this parameter as true, Egress Gateway uses an alternate SCP for routing to an alternate NF destination. If there are fewer SCPs than alternate NF destinations, Egress Gateway tries sending the request through one of the previously used SCPs. **Note**: Additional SCP configuration is required for egress gateway. For more information, see Oracle Communications Cloud Native Core Policy Installation and Upgrade Guide.
Use Binding Information from Binding Header	Specifies whether to enable or disable the use of information received in 3gpp-sbi-binding header. When Policy receives the unique NF Instance ID in the error response, it retrieves the 3gpp-sbi-binding header for that particular NF instance. It extracts the `SetId` to choose the alternate producer. This parameter is considered only for subsequent retry requests. By default, this switch is disabled. **Note**: When this switch is enabled, the value of NFSet Preference is not taken into consideration.
Use Binding Information from NF Profile	Specifies whether to enable or disable the use of information from the set NF Profile. When Policy receives the unique NF Instance ID in the error response, and it does not receive information from the 3gpp-sbi-binding header for that particular NF instance, it extracts the `SetId` to choose the alternate producer. This parameter is considered only for subsequent retry requests. By default, this switch is disabled. **Note**: For subsequent requests, this field takes precedence only if the **Use Binding Information from Binding Header** flag is disabled or no NF binding is received in the initial request.

Table 7-10 (Cont.) Create Retry Profile

Field Name	Description
NF Set Resolution	Specifies whether to enable the on demand discovery of subsequent messages based on the NFSet ID, in case of failure response. The possible values are: • **NRF Discovery:** If this value is selected, then PCF sends a request to NRF to get the list of NFs based on the `SetId` value from the *3gpp-Sbi-Binding header* of the NF with a successful Initial Request. • **Cached NF Profiles:** If this value is selected, then PCF selects the list of NFs from the already discovered NF list. By default **CachedNFProfiles** remains selected. **Note**: NRF discovery is applicable for subsequent messages functionality only. For the other retry requests, the NF list is provided through the cached NF profiles. If the **Cached NF Profiles** option is selected for the subsequent messages, then it uses the DNS-SRV configuration for session retry.
NF Set Preference	Specifies the NF set preference when choosing the alternate producer for an initial request. Policy excludes the producer (received in the failed request) and selects an alternate producer based on the value you choose from the drop-down menu: • Same: Choose the next producer from the same NFSet as the previous request. • Different: Choose the next producer from a different NFSet. • None **(default)**: NFSets filter does not apply. **Note**: This parameter applies only to initial requests. For subsequent requests, the default behavior is to choose the same NFSet provided by Sbi-Binding Info (if enabled). Otherwise, it goes to the same NFSet as the the current NF. In case the current NF does not belong to any NFSet or NFSet is missing, it falls back to DNS-SRV.
Locality Preference	Specifies the locality preference when choosing the alternate producer for a failed request. Policy excludes the producer (received in the failed request) and selects an alternate producer based on the value you choose from the drop-down menu: • Same: Choose the next producer from the same locality as the previous request. • Different: Choose the next producer from a different locality. • None **(default)**: Locality filter does not apply.
Re-attempt to Same NF	Specifies whether Policy shall send the request to the same NF that failed in the first attempt. If you enable this switch, Policy allows re-attempting to the same NF based on the following conditions: • If the server header in error-response has SCP host alone, Policy does not consider error code and error cause, and retries to the same NF. • If the server header in error-response has instances with or without SCP or envoy, Policy checks error code and error cause to find out if retry to same NF is allowed. By default, this switch is disabled. **Note**: Make sure to configure the value for **Re-attempt same NF for following Error Codes** field.

Table 7-10 (Cont.) Create Retry Profile

Field Name	Description
Re-attempt Same NF for following Error Codes	Specifies the HTTP error codes for which Policy does not look for alternate producer and retries the request to the same NF or SCP that failed in the first attempt. To add a value, click **Add** and enter values for the following fields: • **Error Code** - Specifies the HTTP error code. For example, 504 • **Error Causes** - Specifies the error cause. For example, GATEWAY_TIMEOUT, NF_TIMEOUT. You can configure up to 10 error codes with each error code supporting up to five causes. When the field is left empty (default), Policy does not send the request to the NF that failed in the first attempt. **NOTE:** Policy considers this parameter only when **Re-attempt to same NF** switch is enabled.
Enable DNS Resolution	Specifies whether to enable or disable falling back to DNS resolution. When this flag is enabled, during subsequent requests, after the application of filters and flags if no alternate NFs are found, Policy falls back to DNS resolution (DNS SRV). By default, this flag is disabled. **Note**: **Enable FallBack to Higher Priority Destination** and **Retry After Interval** fields are considered only if **Enable DNS Resolution** field is enabled.

Table 7-10 (Cont.) Create Retry Profile

Field Name	Description

Enable Fallback to Higher Priority Destination	Specifies whether to enable or disable falling back to the higher priority destination that failed earlier (after the retry interval). Policy attempts fallback even if the current destination is available.
	When you enable this switch, Policy attempts to fallback to a higher priority NF destination. The priority of NF destination is determined by Alternate Route service (either via DNS-SRV resolution or by static configuration).
	Default Value: False
	The following example illustrates a scenario when Fallback is enabled and **Retry After Interval** is set to 900 seconds:
	`NRF returned profiles - UDR1 (priority-1), UDR2 (priority-2), UDR3 (priority-3),`
	`Alternate Route configuration for UDR1 - UDR1 (priority-1), UDR1.2 (priority-2), UDR1.3 (priority-3),`
	`Retry After Interval - 900 seconds, Fallback is enabled`
	`9:00 AM PCF receives an SM Create request and establishes a session with UDR1`
	`9:05 AM PCF receives an SM Modify request and routes to UDR1, UDR1 fails, PCF reroutes to UDR1.2`
	`9:10 AM PCF receives an SM Modify request and routes to UDR1.2`
	`9:20 AM PCF receives an SM Modify request and routes to UDR1 (fallback)`
	Note: This configuration applies to subsequent message routing only.
	Priority pool must be enabled to use **Enable FallBack to Higher Priority Destination**.
	For multi-pod deployments where subsequent requests for a subscriber can be distributed across different pods, the fallback to higher priority destination will be effective only when there is sufficient traffic. The messages are distributed to all the pods such that they build up respective cache with failed destination info along with failed time stamp. If this cache is not present on a particular pod, the fallback to higher priority will not happen.

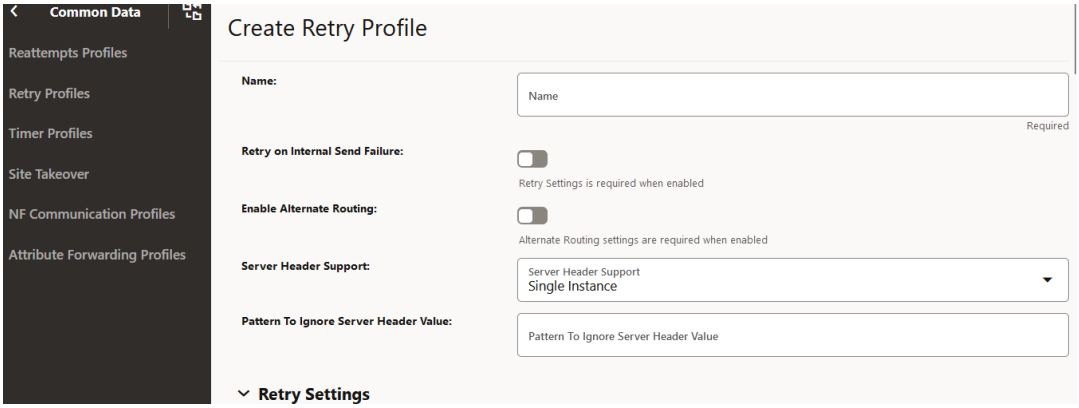
Table 7-10 (Cont.) Create Retry Profile

Field Name	Description

Retry After Interval (in seconds)	Specifies the time in seconds after which Policy re-attempts a request to a higher priority destination.
	In case of fallback, when a route is being determined while the current destination has not failed, PCF attempts to fallback to a higher priority NF destination. Priority of NF destination is determined by Alternate Route service, either through DNS-SRV resolution or static configuration.
	You can set the value for retry after interval from 0 to 86,400 seconds.
	By default, the value is set to 0, which means that a high priority destination is blocklisted.
	Note:
	• This configuration applies to subsequent message routing only.
	• Blacklisting may still be done by Egress Gateway
	• An NF destination (FQDN) is not re-attempted within the context of a single operation, such as SM Modification request
	
	`NRF returned profiles - UDR1 (priority-1), UDR2 (priority-2), UDR3 (priority-3),`
	`Alternate Route configuration for UDR1 - UDR1 (priority-1), UDR1.2 (priority-2), UDR1.3 (priority-3), UDR1.4 (priority-4)`
	`Retry After Interval - 900 seconds, Fallback is disabled`
	
	`9:00 AM PCF receives an SM Create request and establishes a session with UDR1`
	`9:05 AM PCF receives an SM Modify request and routes to UDR1, UDR1 fails, PCF reroutes to UDR1.2`
	`9:10 AM PCF receives an SM Modify request and routes to UDR1.2`
	`9:12 AM PCF receives an SM Modify request and routes to UDR1.2, UDR1.2 fails, PCF reroutes to UDR1.3`
	`9:16 AM PCF receives an SM Modify request and routes to UDR1.3 (no fallback)`
	`9:20 AM PCF receives an SM Modify request and routes to UDR1.3, UDR1.3 fails, PCF alternate routes to UDR1 (not UDR1.4)`

Retry again at end of alternate NF options	When you set the value for this parameter to true, Policy initiates the retry cycle and sends request to the selected NF when the following conditions are true: • Retry attempts made to all available NF instances • Maximum Number of Alternate Routing Attempt is not exhausted

Figure 7-12 Create Retry Profile



4. Click **Save**.
   The retry profile gets listed on the **Retry Profile** page. The page defines the retry profile in the Policy database and it is available to be used in a Policy.

> **Note**
>
> Use or available under the **Actions** column to update or delete the retry profile.

Importing Retry Profiles

To import Retry Profile:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.3.1.3 Timer Profiles

This procedure provides information about how to create and manage timer profiles.

The **Timer Profile** page allows you to create new and manage existing timer profiles. The page displays the list of defined timer profiles with the options to import, export, or add profiles.

To configure timer profile:

1. From the navigation menu under **Policy**, navigate to **Service Configurations**, click **Common Data**, and then select **Timer Profiles**.
   This opens the **Timer Profiles** page. The page lists the existing timer profiles. You can add or import new profiles using this page.

> **Note**
>
> Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Timer Profile** page.
3. Select any of the following services from the **Service Type** drop-down list and perform the required configurations.
   - **PCF User Connector**
     To configure the timer profile for PCF User Connector:
     a. Enter the timeout value for **UDR**:
        Timeout Per Service.nudr-dr (in milliseconds): Specifies the maximum time for the response from the nudr-dr service.

        > **Note**
        > You can enter timeout for specific messages.

     b. Enter the timeout values for **CHF**:
        Timeout Per Service.nchf-spendinglimitcontrol (in milliseconds): Specifies the maximum time for the response from the nchf-spendinglimitcontrol service.
   - **PCF Session Management**
     To configure the timer profile for PCF Session Management:
     a. Enter the timeout values for **SMF**:
        Timeout Per Service.notification (in milliseconds): Specifies the maximum time for the response from the notification service.
     b. Enter the timeout values for **AF**:
        Timeout Per Service.notification (in milliseconds): Specifies the maximum time for the response from the notification service.
   - **PCF Access and Mobility**
     To configure the timer profile for AMF:
     a. Enter the timeout values for **AMF**:
        Timeout Per Service.notification (in milliseconds): Specifies the maximum time for the response from the notification service.
   - **PCF UE Policy**
     To configure the timer profile for PCF UE Policy:
     a. Enter the timeout values for **AMF**:
        Timeout Per Service.namf-comm (in milliseconds): Specifies the maximum time for the response from the namf-comm service.
   - **PCF Binding Service**
     To configure the timer profile for Binding Service
     a. Enter the timeout values for **BSF**:
        Timeout Per Service.nbsf-management (in milliseconds): Specifies the maximum time for the response from the BSF service.
4. Click **Save**.
   The timer profile gets listed on the **Timer Profile** page. The page defines the timer profile in the Policy database and it is available to be used in a Policy.

> **Note**
>
> Use ✎ or 🗑 available under the **Actions** column to update or delete the timer profile.

Importing Timer Profiles

To import Timer Profile:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.3.1.4 Site Takeover

This procedure provides information about how to create and manage the sites for takeovers in case of failure scenarios.

The **Site Takeover** page allows you to create new and manage existing sites for takeover. The page displays the list of defined sites with the options to import, export, or add profiles.

To configure site:

1. From the navigation menu under **Policy**, navigate to **Service Configurations**, click **Common Data**, and then select **Site Takeover**.
   This opens the **Site Takeover** page. The page lists the existing sites. You can add or import new profiles using this page.

> **Note**
>
> Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Site Takeover** page.
3. On the **Create Site Takeover** page, enter the following information:

Table 7-11 Create Site Takeover

Field Name	Description
Name	The unique name for the site. This name is used to refer to the site in other service configuration screen, such as SM Service. The name can only contain the characters A–Z, a–z, 0–9, period (.), hyphen (-), and underline (_). The maximum length is 255 characters.
Site ID	The site ID for which the current site takes control of the pending binding records in case of binding registration failures.

4. Click **Save**.
   The site gets listed on the **Site Takeover** page. The page defines the site in the Policy database and it is available to be used in a Policy.

> **Note**
>
> Use ✎ or 🗑 available under the **Actions** column to update or delete the site.

Importing Site Takeover

To import site:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.3.1.5 NF Communication Profiles

The **NF Communication Profiles** page allows you to create new and manage existing NF communication profiles. The page displays the list of defined profiles with the options to import, export, or add profiles.

To configure NF communications profile, perform the following steps:

1. From the navigation menu under **Policy**, navigate to **Service Configurations**, click **Common Data**, and then select **NF Communication Profiles**.
   This opens the **NF Communication Profiles** page. The page lists the existing profiles. You can add or import new profiles using this page.

> **Note**
>
> Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create NF Communication Profile** page.
3. On the **Create NF Communication Profile** page, enter the following information:

Table 7-12 Create NF Communication Profile

Field Name	Description
NF Communication Profile Name	The unique name for the NF communication profile. This name is used to refer to the profile in other service configuration screen, such as SM Service, User Connector, and so on. The name can only contain the characters A–Z, a–z, 0–9, period (.), hyphen (-), and underline (_). The maximum length is 255 characters.

Table 7-12 (Cont.) Create NF Communication Profile

Field Name	Description
Policy NF Communication Model	Indicates the communication model as defined in 3GPP Technical Specification 23.501 Annexue E, in which the Policy NF is deployed. Select any of the following values from the drop down menu: • **Custom**: Select this option to create a customized communication profile. • **Model B**: Direct communication with NRF interaction. In this model, consumers perform discovery by querying the NRF. Based on the discovery result, the consumer does the selection and sends request to the selected producer. Selecting this option configures the communication profile according to Model B and populates data for the respective input fields. • **Model C**: Indirect communication without delegated discovery. In this model, consumers perform discovery by querying the NRF. Based on the discovery result, the consumer does the selection of an NF set or a specific NF instance of the NF set. The consumer sends the request to the SCP containing the address of the selected service producer pointing to a NF service instance or a set of NF service instances. In the latter case, the SCP selects an NF Service instance. If possible, the SCP interacts with NRF to get selection parameters such as location and capacity. The SCP routes the request to the selected NF service producer instance. Selecting this option configures the communication profile according to Model C and populates data for the respective input fields. • **Model D**: Indirect communication with delegated discovery to SCP. In this model, consumers such as PCF add discovery parameters and delegate the discovery of the producer to SCP. The delegated discovery applies to both on-demand and autonomous discovery scenarios. Selecting this option configures the communication profile according to Model D and populates data for the respective input fields. **Default Value**: Custom
NF Discovery Settings	
Initial Discovery Parameters	Indicates the discovery parameter to be used in the *3gpp-sbi-discovery-<discovery-parameter>* for all Policy Model D NF Communication. This parameter helps SCP to discover the suitable NF for discovery. Policy supports the following discovery parameters for Model D NF Communication: • `supi` • `data-set (may be included for UDR Discovery)` • `preferred-locality` • `dnn (may be included for BSF, SMF Discovery)` • `snssais` • `guami (maybe included for AMF Discovery)` • `group-id-list ( May be included for "UDR","CHF" discovery.)` Policy supports the following discovery paramters for Custom, Model B, and Model C NF Communications: • `target-nf-set-id` • `supi` • `group-id-list` **Note**: The Nbsf interface does not include SUPI as a discovery parameter that is, SUPI is not suppported for BSF interface.
Send Discovery Header in Initial Messages	This switch indicates if PCF must include the *3gpp-sbi-discovery-<discovery-parameter>* parameters in the initial SBI message. By default the switch remains disabled.

Table 7-12 (Cont.) Create NF Communication Profile

Field Name	Description
Send Discovery Header in Subsequent Message	This switch indicates if PCF must include the following *3gpp-sbi-discovery-<discovery-parameter>* parameters in SBI messages: • Parameters to update a subscription, such as PUT messages when unable to fill in a `3gpp-sbi-routing-binding` header. • Parameters to delete a subscription, such as DELETE messages when unable to fill in a `3gpp-sbi-routing-binding` header. By default the switch remains disabled.
Send Target API Root Header in Initial Messages	Indicates if PCF must include `3gpp-sbi-target-apiroot` in initial request messages. If the user selects Model D for Policy NF Communication Model, the value is set to false. For other Policy NF Communication Model values, the default value is true.
On Demand Discovery Caching	
Force Discovery	Indicates whether to cache the NF profiles in NRF client or to skip caching and receive the response directly from NRF. Value of this parameter can be 0 or 1. • When the value of OC-Force-Rediscovery parameter is set to 0, global cache configuration is taken into account. • When the value of OC-Force-Rediscovery parameter is set to 1, global cache configuration is ignored and NF profiles get retrieved directly from NRF while responding to the query from the backend services. **Default value**: 0
Retention Period (in ms)	Indicates the time a record is allowed to stay in database after expiry. This parameter accepts an integer value to indicate the retention period in milliseconds. When retention period is not configured or saved as null, backend will not send this header to NRF connector. **Default value**: 0
NF Bindings Settings	
Binding Level	Indicates the binding level that must to be included in the *3gpp-sbi-binding* header when PCF adds this header in a message to another NF. Select any of the following values from the drop down menu: • NF Set • NF Instance **Default Value**: NF Set
Send Binding Header	Indicates if PCF includes the *3gpp-sbi-binding* header in SBI messages for the subscription creation, modification, or notification requests and responses, as applicable. By default the switch remains enabled.
Send PCF Service Name in Binding Header	Indicates whether to include the service name in the binding header or not. Possible values: • true: Includes `svcname` in the binding header sent towards NFs such as UDR, AMF, or BSF (customized) except for CHF, which contains the 3GPP defined PCF service name. • false: Does not include `svcname` in the binding header. **Default value** : false
Send Routing Binding Header	Indicates if PCF includes the *3gpp-sbi-routing-binding* header in SBI messages for the subscription creation, modification, or notification requests, as applicable. By default the switch remains disabled.

Table 7-12 (Cont.) Create NF Communication Profile

Field Name	Description
Send Callback Header	Indicates if PCF includes the *3gpp-sbi-callback* header in SBI messages for the subscription creation, modification, or notification requests, as applicable. By default the switch remains disabled.
NF Server Settings	
Send Server Header	Indicates if services Policy microservices include server header while sending an error response.
Server Header Error Codes	Specifies the error codes for which service header is generated. The error codes can be from 100 to 999. **Note**: If no error is specified, then Policy sends server header for all error responses.
Retry & Alternate Routing Settings	
Retry Profile for Initial Messages	Specifies the retry or alternate routing options when an SBI message is not delivered successfully in any of the following scenarios: • POST operation to create a subscription • GET or PATCH request not belonging to a subscription
Retry Profile for Subsequent Messages	Specifies the retry or alternate routing options when an SBI message that updates, deletes or notifies a subscription is not delivered successfully.
NF Correlation Settings	
Send Correlation-info header	This specifies if the PCF should send Subscriber identifier as sbi-header-info or not. By default the switch remains disabled.
Allowed Correlation-info Header Generation Type(s)	Specifies that if correlation header is not received from Consumer NFs, PCF should generate the header. The Correlation-Type supported • SUPI • GPSI • (Select either both or none)

> **Note**
>
> If the Retry and Alternate Routing settings are configured on the NF Communication Profile page, these settings take precedence over the alternate routing settings configured on the service configurations page for each service.

4. Click **Save**.
   The NF communication profile gets listed on the **NF Communication Profiles** page. The communication profile also get listed under the respective service configuration on the following pages:
   - PCF Session Management
   - PCF Access and Mobility
   - PCF Policy Authorization
   - PCF UE Policy Service
   - Binding Service
   - PCF User Connector

> **Note**
>
> The NF Communication Profile must be deleted in the service configurations before deleting from this page. As cyclic references are not handled currently.

> **Note**
>
> Use  or  available under the **Actions** column to update or delete the profile.

Importing NF Communication Profiles

To import NF Communication Profile:

1. Click  .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.3.1.6 Attribute Forwarding Profiles

The **Attribute Forwarding Profiles** page allows you to create new and manage existing Attriute Forwarding profiles. The page displays the list of defined profiles with the options to import, export, or add profiles.

For more details on Attribute Forwarding Profiles, see *Attribute Forwarding Profiles* section in Usage Monitoring on Gx Interface.

To configure Attribute Forwarding Profiles profile, perform the following steps:

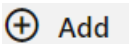
1. From the navigation menu under **Policy**, navigate to **Service Configurations**, click **Common Data**, and then select **Attribute Forwarding Profiles**.
2. Click  .
   This opens the **Create Attribute Forwarding Profiles** page.
3. On the **Create Attribute Forwarding Profiles** page, enter the following information:

Table 7-13 Create Attribute Forwarding Profiles

Field Name	Description
Name	Enter the Attribute Forwarding Profile name.
Forwarded Attributes	Enter the list of attributes to be forwarded from one service to another.

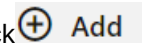
4. To add new Forwarded Attributes, click  .
   This opens the **Add Forwarded Attributes** page.
5. On **Add Forwarded Attributes** page, enter the following details:

Table 7-14 Add Forwarded Attributes

Field Name	Description
Attribute Name	Enter the name of the attribute.
Attribute Source	Specify the source of the attribute to be forwarded. Attribute Source provides the following options: • Request Message: The incoming request message such as Gx CCR message or SM Policy Association Create message. • User Data: Data received from different data sources such as UDR and CHF. • Session Data: Data saved in the session (association) previously, such as saved APN/DNN value.
Attribute Selection	Attribute Selection can be: • Predefined • Custom
Request Message Type	Request Message Type can be: • Diameter Message • HTTP Message **Note**: Currently, Policy supports only Diameter Messages for Request Message Type.
Diameter	This field appears when the Request Message Type is Diameter Message. Specifies the diameter message type such as 3GPP Gx and Gx CC Request.
Attribute Location	CNC Console displays this field when the Attribute Selection is of Custom type. It indicates the path/location at which this attribute can be found by the source (forwarder) micro-service.

6. Click **Save** on **Add Forwarded Attributes** page to add the new Forwarded Attibute.
7. Click **Save** on **Create Attribute Forwarding Profiles** page to add the new Attibute Forwarding Profile.

7.3.2 PCF Session Management

This procedure provides information about configuring the PCF Session Management Service.

The **PCF Session Management** page displays the SM Service configurations. The page allows you to edit the configurations.

To edit PCF Session Management Service:

1. From the navigation menu under **Policy**, navigate to **Service Configurations**, and select **PCF Session Management**.
   This opens the **PCF Session Management** page. The page displays the existing configurations.
2. Click 

.
   This opens the **Edit PCF Session Management** page displaying the default configuration for the fields available in respective groups.

3. Expand the **System** group.
   This group allows you to edit system configurations.
4. Enter the values for the input fields, available under the group.
   The following table describes the fields:

Table 7-15 Edit System Configurations

Field Name	Description
Component Tracing	Determines, if component tracing is enabled. Component tracing is used to evaluate system process latency in detail level. By default, this switch remains disabled.
FQDN	The PCF FQDN used by the PCF to register Binding data to BSF. AF may use this FQDN to communicate with PCF on N5 reference point. FQDN needs to be in a standard FQDN format (RFC 1035). **Default Value**: `pcf-smservice.oracle.com` **Note** : If we have multiple PCF, Diameter identity and FQDN must be unique.
Diameter Realm	The PCF Diameter realm used by the PCF to register Binding data to BSF. Diameter based AF may use this Diameter realm to communicate with PCF on Rx reference point. **Default Value**: `oracle.com`
Diameter Identity	The PCF Diameter identity used by the PCF to register Binding data to BSF. Diameter based AF may use this Diameter identity to communicate with PCF on Rx reference point. **Default Value**: `pcf-smservice.oracle.com` **Note** : In case of multiple PCF, Diameter identity and FQDN must be unique.
Snssai	Used to register Binding data to BSF by PCF. AF or BSF may use this SNSSAI to discover proper PCF. Format is : sst,sd. **Default Value**: `0,000000`
Enable Metrics	This determines if system metrics is enabled. This takes priority on global metrics configuration. By default, this switch remains enabled.
Override Supported Features	Indicates a list of supported features that can be configured to override system embedded values. For example, SessionRuleErrorHandling, NetLoc, PolicyDecisionErrorHandling and so on. **Note:** If any supported feature is enabled in one site then the same needs to be enabled on all other sites at the same time. By default, it is blank.
Enable Custom Json	This determines if custom JSON is enabled. By default, this switch remains disabled.

Table 7-15 (Cont.) Edit System Configurations

Field Name	Description
SMF Notification Retry Profile	Defines the retry profile configuration for session management. For more information on configuring retry profile, see Retry Profiles.
SMF Communication Profile	Specifies the NF communication profile used by SMF. For more information about configuring NF communication profiles, see NF Communication Profiles.
Data Compression Scheme	Specifies the data compression scheme used by SM. Following are the allowed values for this field: • Disabled • MySQL_Compressed • Zlib_Compressed By default, this value remains disabled.

5. Expand the **User** group.
   This group allows you to edit the subscriber configurations.
6. Enter the values for the input fields, available under User group.
   The following table describes the fields:

Table 7-16 Edit User Configurations

Field Name	Description
Validate User	If the User profile, for any of the User Data types given under the **User Data Types** group, are fetched successfully, the user is considered to be a known user. However, if the user profile lookup has failed on the discovered data source, the user profile is marked as Unknown. If this switch is enabled and user is marked as Unknown, the session creation requests are rejected. If this switch is disabled and user is marked as Unknown, the session creation requests are handled and sessions are created successfully.
Query User	Determines if user query from UDR is enabled. When this option is enabled, PCF queries the UDR about the subscriber contained in the SM Association **create** request by sending a GET request for “sm-data” resource on the nudr-dr service. **Note:** The PDS service caches the subscriber profile when `Subscribe To Notify` option is enabled, in that case, the PCF may not always reach the UDR when the subscriber profile is found in the local cache. By default, this switch remains enabled.

Table 7-16 (Cont.) Edit User Configurations

Field Name	Description
Query User on Update	Determines if user query from UDR on update is enabled. When this option is enabled, PDS queries the UDR about the subscriber present in the SM Association **update** request by sending a GET request for `SmPolicyData` resource on the `nudr-dr service`. **Note**: If `Subscribe To Notify` is enabled for SM, then SM will not query the UDR during a Update requests. The PDS caches the subscriber profile when the `Subscribe To Notify` flag is enabled. In that case, Policy may not always reach the UDR when the subscriber profile is found in the local cache. When the `Subscribe To Notify` flag is enabled and previous subscription attempts failed, then along with the data query, PDS also sends a SUBSCRIBE request to UDR. By default, this option is disabled.
Query User on Terminate	Determines if user query from UDR on delete is enabled. When this option is enabled, PCF queries the UDR about the subscriber present in the SM Association **delete** request by sending a GET request for “SmPolicyData” resource on the nudr-dr service. **Note**: If `Subscribe To Notify` is enabled for SM, then SM will not query the UDR during a Terminate requests. The PDS caches the subscriber profile when the “Subscribe To Notify” option is enabled, in that case, the PCF may not always reach the UDR when the subscriber profile is found in the local cache. By default, this option is disabled.
Query User on Reauth	Determines if user query from UDR on reauth is enabled. When this option is enabled, PCF queries the UDR about the subscriber, when it receives a Reauthorization request, such as Rx or Policy Authorization request by sending a GET request for “SmPolicyData” resource on the nudr-dr service. **Note:** The PDS caches the subscriber profile when the “Subscribe To Notify” option is enabled, in that case, the PCF may not always reach the UDR when the subscriber profile is found in the local cache. By default, this option is disabled.
Disable Subscriber Variables	Determines if subscriber variables are stored or not in the PCF database. By default, the value is set to false.
User Data Types	

Table 7-16 (Cont.) Edit User Configurations

Field Name	Description
SmPolicyData	If this switch is enabled, PCF fetches SMPolicyData from nUDR. For information on configuring SMPolicyData attributes, see Table 7-17.
Operator Specific Data	If this switch is enabled, PCF fetches OperatorSpecificData (imported using Custom Schema) from nUDR. For information on configuring SMPolicyData attributes, see Table 7-18.
Ldap Data	If this switch is enabled, PCF fetches user profile from LDAP through PDS.
CHF Data	If this switch is enabled, PCF fetches policy counters from CHF. For information on configuring SMPolicyData attributes, see Table 7-19.
OCS Data	If this switch is enabled, PCF fetches policy counters from OCS. When this switch is enabled, PCF performs on-demand lookup of Policy Counters for a subscriber using the **Fetch Policy Counters** from *OCS* policy action. By default, this option is disabled. **Note**: It is possible to enable both CHF Data and OCS Data simultaneously. Based on PDS-PRE policy, PDS may make a choice between the two datasources. For information on configuring SMPolicyData attributes, see Table 7-20.

The following table describes the **SmPolicyData** switches available in the **Attributes** column:

Table 7-17 SmPolicyData Attributes

Attribute Name	Description
Subscribe to Notify	When this switch is enabled, Policy subscribes with the UDR to get notified on changes in subscriber profile. By default, this option is enabled.
Ignore Subs Notification Check	Ignore subscriber's notification check. By default, this option is false. **Note**: Currently, this field is not used by the Policy application.
Include Snssai in User Query	Determines if Snssai is included for SmPolicyData for PCF query and subscribe to UDR for Notification.
Include Dnn in User Query	Determines if Dnn is included for SmPolicyData for PCF query and subscribe to UDR for Notification.

The following table describes the **Operator Specific Data** switches available in the **Attributes** column:

Table 7-18 Operator Specific Data Attributes

Attribute Name	Description
Subscriber to Notify	When this switch is enabled, Policy subscribes with the UDR to get notified on changes in subscriber profile. By default, this option is false.

The following table describes the **CHF Data** switches available in the **Attributes** column:

Table 7-19 CHF Data Attributes

Attribute Name	Description
Enable Async CHF Query	When this button is enabled, PCF interacts with CHF in Asynchronous mode. By default, this option is disabled.

The following table describes the **OCS Data** switches available in the **Attributes** column:

Table 7-20 OCS Data

Attribute Name	Description
Enable OCS Async Query	When this button is enabled, PCF performs on demand lookup for Policy Counters to be fetched from OCS. By default, this option is disabled.
Enable force lookup on Update	When this button is enabled, PCF performs force lookup for OCS Data. By default, this option is disabled.

7. Expand the **Policy** group.
   This group allows you to edit Policy configurations.
8. Under the **Policy** group, enable or disable the **Evaluate** switch.
   This switch determines, if policy evaluation is enabled.

   The default value is `true`.
9. Expand the **Policy Control Request Trigger** group.
   This group allows you to edit Policy Control Request Trigger configurations.
10. Under the **Policy Control Request Trigger** group, select values in the **Default Policy Control Request Triggers** drop-down list.
    This is the default Policy Control Request Trigger(s) to install on PDU session at SM Policy Association Establishment. You can select multiple values in a comma separated format.
11. Expand the **Binding Configuration** group.
    This group allows you to edit the Binding configurations.
12. Enter the values for the input fields, available under the group.
    The following table describes the fields:

Table 7-21 Edit Binding Configurations

Field Name	Description
Binding Operation	Determines if binding operation (register and deregister) to the BSF is enabled. This service level global configuration applies to all session creation requests (belonging to all DNN and/or S-NSSAI. **Default Value**: TRUE After performing necessary configurations on this screen, user may use the policy action - **Set Binding Registration to** to update configurations for specific policies. For more information, see *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.
Binding Use Local Configured Bsf Always	Whether to use local configured BSF without Always discovering. **Default Value**: FALSE
Binding Use Local Configured Bsf When Not Discovered	Whether to use local configured (if having) BSF when not discovered or discover failed. Local configuration can be done using custom yaml. **Default Value**: FALSE
Use HTTP2	Determines if using http/2 to communicate with BSF. Otherwise use http/1.1. **Default Value** : TRUE
Abort or Terminate Session on Binding Error	Determines if PCF (SM service) should abort terminate session when binding error is received.
BSF Retry Profile for Initial Messages	Retry Profile to be used when PCF fails to send a create message to a producer node. For more information, see Retry Profiles. **Note:** This is a mandatory configuration to get the error codes from the **Attempt Alternate Route for Following Error Codes** option on the Retry Profiles page.

Table 7-21 (Cont.) Edit Binding Configurations

Field Name	Description
BSF Retry Profile for Subsequent Messages	Retry Profile to be used when PCF fails to send an in-session message to a producer node. **Note:** • If both “Retry Profile for Initial Messages” and “Retry Profile for Subsequent Messages” are not configured (default case) no retry is attempted. • If both “Retry Profile for Initial Messages” and “Retry Profile for Subsequent Messages” are configured, retries is attempted accordingly. • If “Retry Profile for Initial Messages” is configured but “Retry Profile for Subsequent Messages” is not configured then the profile for initial messages is used for subsequent messages. (to be backward compatible with release 1.8.x) • If “Retry Profile for Initial Messages” is not configured but “Retry Profile for Subsequent Messages” is configured then retry is not attempted for initial messages but is attempted for subsequent messages.
Default Binding Operation Mode	Allows users to configure binding operation mode as synchronous or asynchronous. **Default value:** Synchronous **Note:** • The mode chosen during the CREATE request remains same during UPDATE and TERMINATE operations for the same SM session. • To ensure backward compatibility, if this attribute is not available in the database, the value is set to "Synchronous" After performing necessary configurations on this screen, user may use the policy action - **Set Binding Registration to** under **PCF-SM Category** to update configurations for specific policies. For more information, see *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.
BSF Communication Profile	Specifies the NF communication profile used by BSF. For more information about configuring NF communication profiles, see NF Communication Profiles.

Table 7-21 (Cont.) Edit Binding Configurations

Field Name	Description
BSF Binding Recreate Attempt	The switch to enable the BSF recreate attempts functionality. Possible values: • **Enabled:** The SM Service creates and updates the PendingOperations if the response from BSF applies for a reattempt and it updates the PendingOperation table in Audit Service. Audit Service sends notifications for retry timestamps that have been reached and SM Service triggers the Recreate attempt. • **Disabled:** SM Service deregisters the PendingOperation table in Audit Service and it does not create nor update entries in the PendingOperation table. • **Disable and Cleanup:** SM Service neither deregister the PendingOperation table in Audit Service nor it updates the table. Audit Service sends notifications to SM Service but the notifications does not get processed and SM service automatically deletes records from the PendingOperation table.
BSF Binding Recreate Attempt Profile	The Reattempt profile to be used for binding recreate attempts. For more information, see Reattempts Profile.

13. Expand the **QoS** group.
This group allows you to edit the QoS configurations.
14. Enter the values for the input fields, available under the group.
The following table describes the fields:

Table 7-22 Edit QoS Configurations

Field Name	Description
Qos Data Id Prefix	This is the prefix of qos data id used by PCF to generate qos data id. For example, prefix is "qosdata_", the generated qos data id is qosdata_0. **Default Value** : qosdata_
update Default Pcf Rule With Auth Def Qos	This determines whether to update Qos of default PccRule with the authDefQos of session rule. **Default Value** : TRUE
Install Default Qos If Not Requested	This determines whether to install default Qos to the PDU session if UE not requested. **Default Value** : TRUE
Default Qos 5qi	This is the 5Qi of default Qos which is applied if no default Qos is requested by UE. **Default Value**: 9

Table 7-22 (Cont.) Edit QoS Configurations

Field Name	Description
Default Qos Arp Preempt Cap	This is the ARP Preemption Capability of default Qos which is applied if no default Qos is requested by UE. **Default Value** : MAY_PREEMPT
Default Qos Arp Preempt Vuln	This is the ARP PreemptionVulnerability of default Qos which is applied if no default Qos is requested by UE. **Default Value** : NOT_PREEMPTABLE
Default Qos Arp Priority Level	This is the ARP Priority Level of default Qos which is applied if no default Qos is requested by UE. **Default Value**: 1

15. Expand the **Rule** group.
    This group allows you to edit the PCC Rule configurations.
16. Enter the values for the input fields, available under the group.
    The following table describes the fields:

Table 7-23 Edit Rule Configurations

Field Name	Description
Install Default Pcc Rule	This determines whether and how to install default pcc rule for a PDU session. Possible values are: • ALWAYS • IF_NO_PROVISIONED_RULE only if no other provisioned rule is configured • IF_NO_RULE only if no other rule (predefined or provisioned) is configured/ installed • NEVER **Default Value** : IF_NO_RULE
Default PCC Rule Profile	(Optional) If a default PCC Rule Profile is configured by the user, it is applied to the chosen PCC Rule.
Rule Id Prefix	This is the prefix of rule id of the pcc rule or session rule auto generated by PCF. for example, prefix is "0_", the generated rule id is 0_0, 0_1, etc.
Default Pcc Rule 5qi	This is the 5Qi of default pcc rule. **Default Value**: 9
Default Pcc Rule Precedence	This is the precedence of default pcc rule. **Default Value** : 3000
Default Pcc Rule Arp Preempt Cap	This is the ARP Preemption Capability of qos of default PCC rule. **Default Value** : NOT_PREEMPT
Default Pcc Rule Arp Preempt Vuln	This is the ARP PreemptionVulnerability of qos of default pcc rule. **Default Value** : PREEMPTABLE

Table 7-23 (Cont.) Edit Rule Configurations

Field Name	Description
Default PCC Rule Requested Rule Data	Used to set the default Requested Rule Data for the PCC Rule. **Default Value**: SUCC_RES_ALLO
App Rule Precedence Min	This value defines the minimum value for precedence of a PCC rule as authorized by the establishment of an application flow by the AF. If multiple rules are applied to the same packet flow or UE resource (i.e., overlapping rules) a rule with lower precedence value takes the priority over a rule with higher precedence value. The value of -1 is used to not set the precedence of a rule (NOT RECOMMENDED). **Default Value**: 400
App Rule Precedence Max	This value defines the maximum value for precedence of a PCC rule as authorized by the establishment of an application flow by the AF. If multiple rules are applied to the same packet flow or UE resource (i.e., overlapping rules) a rule with lower precedence value takes the priority over a rule with higher precedence value. The value of -1 is used to not set the precedence of a rule (NOT RECOMMENDED). **Default Value**: 899
Default Pcc Rule Arp Priority Level	This is the ARP Priority Level of qos of default pcc rule The range is 1 to 15. Values are ordered in decreasing order of priority, for example, with 1 as the highest priority and 15 as the lowest priority. **Default Value** : 15
Switch Flow In To Out Enabled	This determines whether to switch "in" to "out" in flow description. The src and desc is switched as well. For example, if enabled, "permit in ip from 2800:a00:cc01:c056:1c00:de10:c481:f193/128 to 2800:a00:800:7::1:3b/128 36004" is changed to "permit out ip from 2800:a00:800:7::1:3b/128 36004 to 2800:a00:cc01:c056:1c00:de10:c481:f193/128" **Default Value**: FALSE
Set PacketFilterUsage to true for Preliminary Service Info	This determines whether the UE shall be provisioned with the packet filter for preliminary service.**Default Value**: FALSE

Table 7-23 (Cont.) Edit Rule Configurations

Field Name	Description
Install/Remove Rule Conflicts Strategy	When the remove action is **Remove ALL**(ALL, Predefined, Dynamic, Conditioned, non-conditioned): • INSTALL/MODIFY: SM Service removes all Session/PCC Rules previously installed and ignores all the remove actions for rules in conflict. • REMOVE: SM Service removes all Session/PCC Rules previously installed and ignores all the install actions for rules in conflict. • IGNORE: SM Service removes all Session/Pcc Rules previously installed and ignores all actions for rules in conflict, and does not run anything(install/remove). • Default: SM Service processes the remove actions and then the INSTALL or MODIFY actions.

17. Expand the **Charging** group.
This group allows you to edit the charging configurations.
18. Enter the values for the input fields, available under the group.
The following table describes the fields:

Table 7-24 Edit Charging Configurations

Field Name	Description
Charging Data Id Prefix	This is the prefix of chg data id used by PCF to generate chg data id. For example, prefix is "chgdata_", the generated chg data id is chgdata_0, chgdata_1, etc. **Default Value**: chgdata_
Primary CHF Address	Address of the primary CHF
Primary CHF Instance Id	Instance ID of the primary CHF
Primary CHF Set Id	Set ID of the primary CHF
Secondary CHF Address	Address of the secondary CHF
Secondary CHF Instance Id	Instance ID of the secondary CHF
Secondary CHF Set Id	Set ID of the secondary CHF
Online	Indicates the online charging is applicable to the PDU session.
Offline	Indicates the offline charging is applicable to the PDU session.
Use Subscriber Default Charging Method	Indicates whether to use the online value defined in the PDU session. **Default Value**: true

19. Expand the **Traffic Control** group.
This group allows you to edit Traffic Control configurations.
20. Under the **Traffic Control** group, enter value of the **Traffic Control Id Prefix** field.
This is the prefix of traffic control data ID used by PCF to generate tc data id. For example, if prefix is "tcdata_", the generated tc data ID is tcdata_0, tcdata_1, and so on.

Default Value: tcdata_

21. Expand the **IMS Emergency Session** group.
    This group allows you to edit IMS Emergency Session configurations.
22. Enter the values for the input fields, available under the group.
    The following table describes the fields:

Table 7-25 Edit IMS Emergency Session Configurations

Field Name	Description
Emergency DNNs	DNN included in the request from SMF to PCF to establish an emergency session.
Priority Level	Defines the relative importance of a resource request. **Default Value**: 1
Preemption Capability	Defines whether a service data flow may get resources that were already assigned to another service data flow with a lower priority level. **Default Value**: MAY_PREEMPT
Preemption Vulnerability	Defines whether a service data flow may lose the resources assigned to it in order to admit a service data flow with higher priority level. **Default Value**: NOT_PREEMPTABLE

23. Expand the **Audit** group.
    This group allows you to edit Audit configurations for **SM Policy Association** and **Pending Operation** DB tables.
24. Enter the values for the input fields, available under the **SM Policy Association** group.
    The following table describes the **SM Policy Association** fields:

Table 7-26 Edit SM Policy Association Configurations

Field Name	Description
Enabled	Determines whether to send registration request to Audit service or not. **Default Value**: True
Notification Rate (per second)	Defines the number of stale records which Audit service notifies to Session Management (SM) service in one second. **Default and Recommended Value**: 50 **Note**: To configure higher number than the recommended value, contact My Oracle Support (https://support.oracle.com)
Policy Association Age (in minutes)	Defines the age of a SM policy association after which a record is considered to be stale on PCF and the SMF is queried for presence of such associations. **Default Value**: 1440
Policy Association Maximum Age (in minutes)	Defines the maximum age of a SM policy association after which a record is purged from PCF SM database without sending further queries to SM. **Default Value**: 2880

Table 7-26 (Cont.) Edit SM Policy Association Configurations

Field Name	Description
Minimum Audit Attempts	Specifies the minimum number of consecutive failed audit attempts until maxTTL / forceTTL is reached. If maxTTL is reached and audit_attempts + 1 >= Minimum Audit Attempts for maxTTL, Audit service sends notification to SM service with maxTTL flag set to *true*. SM Service sends DELETE request to PDS and Binding Service. **Range**: 0-99 **Default Value**: 0 **Note**: If maxTTL is not reached and if audit attempts are reached, the number of audit attempts are incremented until maxTTL is reached.
Minimum Audit Passes Interval (in minutes)	Defines the time when next audit for the SM service table is done after delta time if auditing this table has been finished before this specified time. **Default Value**: 330

25. Enter the values for the input fields, available under the **Pending Operation** group. The following table describes the **Pending Operation** fields:

Table 7-27 Edit Pending Operation Configurations

Field Name	Description
Notification Rate (per second)	Defines the number of pending operation records which Audit service notifies to Session Management (SM) service in one second. **Note:** The notification rate for all the operations of Audit service remains the same. Policy does not allow to configure different values for different operations. Therefore, by default the notification rate configured for the **SmPolicyAssociation** section applies for **Pending Operations**. **Default Value**: 50
Pending Operation Age (in minutes)	Defines the age of a pending operation after which a record is considered to be stale. This is a non modifiable field. **Default Value**: 0
Pending Operation Maximum Age	Defines the maximum age of a pending operation after which a record is purged from PCF SM database without sending further queries to SM. **Default Value**: 2880
Minimum Audit Passes Interval (in minutes)	Defines the frequency with which the Audit service performs audit of the pending operation table for the Retry Timestamps that have been reached. **Default Value**: 10

Table 7-27 (Cont.) Edit Pending Operation Configurations

Field Name	Description
Alternate Site	The site takeover profile. For more information, see Site Takeover.
Enable Alternate Site	Specifies if the site takeover must be enabled in case of failure.

26. Enter the values for the input fields, available under the **Pending Transaction** group. The following table describes the fields:

Table 7-28 Edit Pending Transaction Configurations

Field Name	Description
UpdateNotify Retry Backoff (in milliseconds)	Indicates the time between two consecutive retry UpdateNotify messages when UpdateNotify message is held back from being sent to SMF. The default value for this field is 1000.
UpdateNotify Retry Count	Indicates the count for UpdateNotify message to be retried when it is held back from sending to SMF. The default value for this field is 2.
Error Update Notify Handling Rx	On Update Notify to SMF triggered by Rx, this configuration decides if it should RETRY (with UpdateNotifyRetryBackoff and UpdateNotifyRetryCount settings) or DISCARD when receiving 400 Pending Transaction response from SMF. The default value for this field is DISCARD. The allowed values are: • DISCARD (0) • RETRY (1)
Error Update Notify Handling Subscription Notification	On Update Notify to SMF triggered by Subscriber Notification, this configuration decides if it should RETRY (with UpdateNotifyRetryBackoff and UpdateNotifyRetryCount settings) or DISCARD when receiving 400 Pending Transaction response from SMF. The default value for this field is RETRY. The allowed values are: • DISCARD (0) • RETRY (1)

27. Under the **NWDAF** group, enable or disable the **Enable NWDAF Data** switch.
28. Perform the following steps to configure **Advanced Settings** that allows PCF to handle concurrent requests:
    a. Click the ⊕ Add
    The page opens the **Add Advanced Settings** dialog box.
    b. In the dialog box, enter the following **keys** and respective **values**:

    The following table describes the keys and values:

Table 7-29 Add Advanced Settings Configurations

Key	Value
CONCURRENCY.LOCK_LEASE_DURATION_FOR_CREATE	The duration for which lock is kept once the acquisition is successful. After this duration, the lock gets released automatically.**Default value:** 6 seconds
CONCURRENCY.LOCK_WAIT_DURATION_FOR_CREATE	Duration by which the SM service waits for the response to get a lock.
CONCURRENCY.LOCK_REQUEST_RETRY_COUNT_FOR_CREATE	Number of attempts the service make to request the lock.
CONCURRENCY.BULWARK_ENABLED_FOR_N7_CREATE	Enables concurrency for create request. **Default value:** False
CONCURRENCY.LOCK_REQUEST_RETRY_BACKOFF	Defines the amount of time after which the service retries to gain the lock, incase of failure. **Default value:** 750ms
CONCURRENCY.LOCK_LEASE_DURATION_FOR_DELETE	The duration for which lock is kept once the acquisition is successful. After this duration, the lock gets released automatically.**Default value:** 6 seconds
CONCURRENCY.LOCK_WAIT_DURATION_FOR_N7_DELETE	Duration by which the SM service waits for the response to get a lock. **Default value:** 3 seconds
CONCURRENCY.LOCK_REQUEST_RETRY_COUNT_FOR_DELETE	Number of attempts the service make to request the lock. **Default value:** 3 seconds
CONCURRENCY.BULWARK_ENABLED_FOR_N7_DELETE	Specifies if the concurrency has to be enabled for the SM Delete requests. **Default value:** False
CONCURRENCY.LOCK_LEASE_DURATION_FOR_UPDATE	The duration for which lock is kept once the acquisition is successful. After this duration, the lock gets released automatically.**Default value:** 6 seconds
CONCURRENCY.LOCK_WAIT_DURATION_FOR_UPDATE	Duration by which the SM service waits for the response to get a lock. **Default value:** 3 seconds
CONCURRENCY.LOCK_REQUEST_RETRY_COUNT_FOR_UPDATE	Number of attempts the service can make to request the lock. **Default value:** 3
CONCURRENCY.BULWARK_ENABLED_FOR_N7_UPDATE	Specifies if the concurrency has to be enabled for the SM Update requests. **Default value:** False
CONCURRENCY.LOCK_LEASE_DURATION_FOR_CLEANUP	The duration for which lock is kept once the acquisition is successful. After this duration, the lock gets released automatically.**Default value:** 6 seconds
CONCURRENCY.LOCK_WAIT_DURATION_FOR_CLEANUP	Duration by which the SM service waits for the response to get a lock. **Default value:** 3 seconds
CONCURRENCY.LOCK_REQUEST_RETRY_COUNT_FOR_CLEANUP	Number of attempts the service can make to request the lock. **Default value:** 3

Table 7-29 (Cont.) Add Advanced Settings Configurations

Key	Value
CONCURRENCY.BULWARK_ENABLED_FOR_N7_CLEANUP	Specifies if the concurrency has to be enabled for the SM Clean requests. **Default value:** False
RESOURCEID.SUFFIX_LIST	Select a subscriber value type from the drop down for which the lock must be requested: • SUPI • GPSI
CONCURRENCY.LOCK_REQUEST_RETRY_BACKOFF	Defines the amount of time after which the service retries to gain the lock, incase of failure. **Default value:** 750ms
BINDING-BSF_REATTEMPT_THRESHOLD	Defines the frequency with which SM service calls the Audit service API for checking the **Pending Operation** threshold.
USER.RESET_CONTEXT_SM_POLICY_DATA_ON_SM_CREATE	If this configuration is enabled, the sm-service on receiving the sm-create for a SUPI, checks if there is no other `smPolicyAssociation` for the same SUPI and sets `resetContext` flag for the SM policy on the request towards PDS. **Default Value**: False
USER.RESET_CONTEXT_OSD_ON_SM_CREATE	If this configuration is enabled, the sm-service on receiving the sm-create for a SUPI, checks if there is no other `smPolicyAssociation` for the same SUPI and sets `resetContext` flag for the OSD on the request towards PDS. **Default Value**: False
USER.RESET_CONTEXT_CHF_DATA_ON_SM_CREATE	If this configuration is enabled, the sm-service on receiving the sm-create for a SUPI, checks if there is no other `smPolicyAssociation` for the same SUPI and sets `resetContext` flag for the CHF on the request towards PDS. **Default Value**: False
USER.RESET_CONTEXT_LDAP_DATA_ON_SM_CREATE	If this configuration is enabled, the sm-service on receiving the sm-create for a SUPI, checks if there is no other `smPolicyAssociation` for the same SUPI and sets `resetContext` flag for the LDAP on the request towards PDS. **Default Value**: False
USER.RESET_CONTEXT_SSV_ON_SM_CREATE	If this configuration is enabled, the sm-service on receiving the sm-create for a SUPI, checks if there is no other `smPolicyAssociation` for the same SUPI and sets `resetContext flag` for the SSV on the request towards PDS. **Default Value**: False
CONCURRENCY.LOCK_LEASE_DURATION_FOR_UPDATE_NOTIFY	Duration for which lock is held once it is acquired. After this duration, the lock will be released automatically. **Default Value**: 6 Seconds

Table 7-29 (Cont.) Add Advanced Settings Configurations

Key	Value
CONCURRENCY.LOCK_WAIT_DURATION_FOR_UPDATE_NOTIFY	Duration for which the Policy service will wait for the response to get a lock. The same duration is used by Bulwark service to poll for the lock if a lock is not available. **Default Value**: 3 Seconds
CONCURRENCY.LOCK_REQUEST_RETRY_COUNT_FOR_UPDATE_NOTIFY	The number of retries when a request to Bulwark service fails. **Default Value**: 3
CONCURRENCY.BULWARK_ENABLED_FOR_N7_UPDATE_NOTIFY	Determines whether to enable Bulwark service for SM notification flow. **Default Value**: false
CONCURRENCY.UPDATE_USERDATA_IN_ASSOCIATION_ON_LOCKFAILURE	Facilitates the user to update user data in the database in the case of lock failure. **Default Value**: false
NWDAF.EVENT	Allow to add one or more Events to be send in the request towards NWDAF. **Default value**: SLICE_LOAD_LEVEL
NWDAF.REQUEST_TIMEOUT	Allow to control the timeout of the request from SM towards NWDAF. **Default value**: 3000 (ms)
SYSTEM.PA_MAX_APP_SESSIONS	Specifies the limit on the number of AppSessions per SM session. **Default value**: 4
USER.excludeDnnSet.<dnnSetId>	List of Data Network Names (DNNs) for which all the requests or sessions will be excluded for all the User Data Types. The DNN list can include the comma separated values. Regular expressions (regex) are allowed for the DNN names with the following conditions: • must not contain white spaces. • must be case insensitive. • can not be an empty string. • '\' (backslash) must be specified as '/d'. • the names must be separated only with commas (,).
USER.smPolicyData.excludeDnn.<Snssai>	Specify the Snssai corresponding to the DNNs for which all the requests or sessions will be excluded for smPolicyData. Snssai corresponding to the DNN must be the dnnSetId configured in the key from `USER.excludeDnnSet.<dnnSetId>`. There is no regex applied to the snssai value.
USER.ldapData.excludeDnn.<Snssai>	Specify the Snssai corresponding to the DNNs for which all the requests or sessions will be excluded for the given Ldap Data.
USER.operatorSpecificData.excludeDnn.<Snssai>	Specify the Snssai corresponding to the DNNs for which all the requests or sessions will be excluded for the given Operator Specific Data.

Table 7-29 (Cont.) Add Advanced Settings Configurations

Key	Value
USER.chfData.excludeDnn.<Snssai>	Specify the Snssai corresponding to the DNNs for which all the requests or sessions will be excluded for the given CHF Data.
USER.ocsData.excludeDnn.<Snssai>	Specify the Snssai corresponding to the DNNs for which all the requests or sessions will be excluded for the given OCS Data.
USER.ssv.excludeDnn.<Snssai>	Specify the Snssai corresponding to the DNNs for which all the requests or sessions are excluded for the given SSV Data. Example: USER.ssv.excludeDnn.snssai1 = dnnSet1
USER.CREATE_CONTEXT_ON_FAILURE_OCS_DATA	When the value of this key is true, PCRF core sends this flag as part request parameters to PDS. PDS creates a dummy context for OCS in case of getting error while performing OCS lookup. This will support the PDS when it receives notificaiton from UDR to fetch OCS data. This is applicable for SM Create and SM Update. This is applicable for SmCreate and SmUpdate with Asynchronous flow either set as true or false. **Default Value**: false
SYSTEM.SMF_MAX_WAIT_DURATION_FOR_DELETE	Specify the wait time for SMF to trigger delete request on receiving successful response for terminateNotify request. The allowed value range from 0 to 30000 milliseconds. **Note**: If value is either less than 0ms or more than 30000ms, by default 30000ms is considered. The default value for this parameter is 30000ms.
USE_TGPP_SBI_MSG_PRIORITY	Specify the use of `3gpp-sbi-message-priority` header for lock requests. Allowed values are `true` or `false`. **Default value**: true
REQUEST_PRIORITY_FOR_SM_DELETE	Specify the setting request priority for delete requests. Allowed integer number ranging from 0-100. **Default value**: 16
REQUEST_PRIORITY_FOR_SM_UPDATE	Specify the setting request priority for update requests. Allowed integer number ranging from 0-100. **Default value**: 18
REQUEST_PRIORITY_FOR_SM_UPDATE_NOTIFY	Specify the setting request priority for update notify requests. Allowed integer number ranging from 0-100. **Default value**: 18
REQUEST_PRIORITY_FOR_SM_CLEANUP	Specify the setting request priority for cleanup requests. Allowed integer number ranging from 0-100. **Default value**: 18

Table 7-29 (Cont.) Add Advanced Settings Configurations

Key	Value
REQUEST_PRIORITY_FOR_SM_CREATE	Specify the setting request priority for cleanup requests. Allowed integer number ranging from 0-100. **Default value**: 24
ENABLE_SM_POLICY_ASSOCIATION_TABLE_SLICING	Indicates whether to enable/disable the database operations on sliced tables. That is, whenever ENABLE_SM_POLICY_ASSOCIATION_TABLE_SLICING flag is set to `true`, the create/retrieve/update/delete operations are performed on the database slices instead of the main database. **Default value**: false
AUDIT_SMPOLICY_ASSOCIATION.MAX_TTL.TERMINATE_NOTIFY.ENABLED	When SM service receives a MaxTTL notification from Audit service, and if the value of this advanced settings key is set to `true`, SM service sends Terminate Notify to SMF. **Default value**: false
SYSTEM.COLLISION_DETECTION.TERMINATE_NOTIFY.ENABLED	When a collision is detected and the existing session is older than the new one, and if the value of this advanced settings key is set to `true`, SM service sends TerminateNotify to SMF. **Default value**: false
AUDIT.HTTP2_ENABLED	Determines whether to use http/2 or http/1.1 to communicate with Audit service for threshold polling task. Possible values: • true: http/2 is used to communicate with Audit service • false: http/1.1 is used to communicate with Audit service **Default value**: true
USER.CREATE_CONTEXT_ON_FAILURE_SM_POLICY_DATA	If this flag is enabled the SM service sends this as a request parameter toward PDS. At PDS, if it receives an error during SmPolicyData-UDR lookup, then PDS creates a dummy context for SmPolicyData. This creation of dummy context helps, when PDS receives next request for a UDR lookup and this time it is successful, then both the sessions will have the latest SmPolicyData. This is applicable for SmCreate and SmUpdate. **Default value**: false

Table 7-29 (Cont.) Add Advanced Settings Configurations

Key	Value
USER.CREATE_CONTEXT_ON_FAILURE_CHF_DATA	If this flag is enabled the SM service sends this as a request parameter toward PDS. PDS creates a dummy context for CHF in case of getting error while performing CHF lookup. This creation of dummy context helps, when PDS receive next request for a CHF lookup and this time it is successful, then both the sessions will have the latest information. This is applicable for SmCreate and SmUpdate with Asynchronous flow either set as true or false. **Default value**: false
USER.CREATE_CONTEXT_ON_FAILURE_OPERATOR_SPECIFIC_DATA	If this flag is enabled the SM service sends this as a request parameter toward PDS. At PDS, if it receives an error during operatorSpecificData-UDR lookup, then PDS creates a dummy context for operatorSpecificData. This creation of dummy context helps, when PDS receives next request for a UDR lookup and this time it is successful, then both the sessions will have the latest operatorSpecificData. This is applicable for SmCreate and SmUpdate. **Default value**: false
RULE.ENABLE_PCC_RULE_REMOVE_ON_FAILURE	If this flag is enabled and the update notify request fails for PCC rule remove with any HTTP error, then the association will be updated with rule removed. **Default value**: false
USER.allDataTypes.excludeDnns	Specifies the comma separated list of DNNs. For any request that matches one of the DNNs in the list, the communication with PDS will be excluded. Regular expressions (regex) are applied as supported by regex from java Pattern class with the following special cases: • All whitespaces should be removed before processing any values in the arrays. • Every dnn should contain regex characters '^' at the start and '$' at the end. The dnn can not be an empty string. • dnn exclude settings are case insensitive • For security reasons '\' (backslash) is not supported, to interpret regex conditions like '\d' it will need to be written as '/d',. Internally all '/' will be translated to '' for java to handle backslash regex conditions. • { n, m } repetitions are not supported as ',' will be used to separate each dnn.

Table 7-29 (Cont.) Add Advanced Settings Configurations

Key	Value
BINDING.excludeDnns	Specifies the comma separated list of DNNs. For any request that matches one of the DNNs in the list, the communication with Binding Service will be excluded. Regular expressions (regex) are applied as supported by regex from java Pattern class with the following special cases: • All whitespaces should be removed before processing any values in the arrays. • Every dnn should contain regex characters '^' at the start and '$' at the end. The dnn can not be an empty string. • dnn exclude settings are case insensitive • For security reasons '\' (backslash) is not supported, to interpret regex conditions like '\d' it will need to be written as '/d',. Internally all '/' will be translated to '' for java to handle backslash regex conditions. • { n, m } repetitions are not supported as ',' will be used to separate each dnn.
BULWARK.excludeDnns	Specifies the comma separated list of DNNs. For any request that matches one of the DNNs in the list, the communication with Bulwark Service will be excluded. Regular expressions (regex) are applied as supported by regex from java Pattern class with the following special cases: • All whitespaces should be removed before processing any values in the arrays. • Every dnn should contain regex characters '^' at the start and '$' at the end. The dnn can not be an empty string. • dnn exclude settings are case insensitive • For security reasons '\' (backslash) is not supported, to interpret regex conditions like '\d' it will need to be written as '/d',. Internally all '/' will be translated to '' for java to handle backslash regex conditions. • { n, m } repetitions are not supported as ',' will be used to separate each dnn.

Table 7-29 (Cont.) Add Advanced Settings Configurations

Key	Value
CONCURRENCY.LOCK_RETRY_MODE	Indicates whether the lock retry must be triggered under SM service. Possible valures are: • **CLIENT_ONLY**: lock retry is triggered only in client services. • **SERVER_ONLY**: lock retry is triggered only in bulwark. • **CLIENT_AND_SERVER**: lock retry is triggered by both server and client.(server will retry to acquire the lock till the lockWaitTimeout is expired and client will retry after the configured request backOff timer is expired) • Consumer services's advance settings should be updated with the non-zero lockWaitTimeOut.
SYSTEM.ENABLE_CLEANUP_DELAY_AFTER_FAILED_REQUEST	During SM create, when request is rejected or there is an exception, this flag lets the user decide if unsubscribe request to UDR and CHF should be sent immediately or after a certain delay. **Default value**: false
SYSTEM.CLEANUP_DELAY_AFTER_FAILED_REQUEST	When delay on unsubscribe request after failure flag is enabled, this flag lets the user configure the required delay before sending unsubscribe request to UDR and CHF. The value of this flag is in miliseconds. **Default value**: 2000
SYSTEM.ENABLE_APP_SESSION_INFO_CLEANUP_ON_DATA_TRUNCATION	If this flag is enabled on SMUpdate or SmUpdateNotify flows while saving the SmPolicyAssociation and SQL DataTruncation exception happens, then all of the old AppSessions related to the SmPolicyAssociation which are beyond the SYSTEM.PA_MAX_APP_SESSIONS limit will be cleaned up in X ms intervals. Here, X can be configured by the user. After cleaning up the AppSessionInfo and related pcc rules, SMF Update Notify is sent if required and the available corresponding AppSessions from DB is deleted. **Default value**: false
SYSTEM.APP_SESSION_INFO_CLEANUP_DELAY_ON_DATA_TRUNCATION	The time interval between each AppSessionInfo cleanup execution when SYSTEM.ENABLE_APP_SESSION_INFO_CLEANUP_ON_DATA_TRUNCATION takes action. It is defined in milliseconds. **Default value**: 1000

Table 7-29 (Cont.) Add Advanced Settings Configurations

Key	Value
RULE.SET_PACKETFILTERUSAGE_TRUE_FOR_FINALSERVICEINFO	This configuration flag allows the user to enable the *packetFilterUsage* field to be sent with `true` during the update notification to SMF when the main request is set as FINAL for `servInfoStatus` field. If the main request does not specify a serviceInfoStatus value, SM sets this field with value as FINAL. **Default value**: false

Table 7-30 Add Advanced Settings Configurations

Key	Value
SM.UPDATE.EVENT.SUBS.PRIORITY	Set request priority for creation or modification of an event subscription request. **Allowed Values**: 0-31 **Default Value**: 18
SM.CREATE.PRIORITY	Set request priority for SM create request. **Allowed Values**: 0-31 **Default Value**: 24
SM.SUB.FAIL.NOTIFY.PRIORITY	Set request priority for UDR subscription failure notification request. **Allowed Values**: 0-31 **Default Value**: 26
SM.USER.SERVICE.NOTIFY.PRIORITY	Set request priority for User Data change notification request. **Allowed Values**: 0-31 **Default Value**: 18
SM.UPDATE_PRIORITY	Set request priority for SM update request. **Allowed Values**: 0-31 **Default Value**: 18
SM.REAUTH_PRIORITY	Set request priority for SM reauthorization request. **Allowed Values**: 0-31 **Default Value**: 20
SM.DELETE.PRIORITY	Set request priority for SM delete request. **Allowed Values**: 0-31 **Default Value**: 16
SM.POLICY.CLEANUP.PRIORITY	Set request priority for clean up SM policies request. **Allowed Values**: 0-31 **Default Value**: 20
SM.APP.SESSION.CREATE.PRIORITY	Set request priority for Policy authorization create request. **Allowed Values**: 0-31 **Default Value**: 24

Table 7-30 (Cont.) Add Advanced Settings Configurations

Key	Value
SM.APP.SESSION.DELETE.PRIORITY	Set request priority for Policy authorization delete request. **Allowed Values**: 0-31 **Default Value**: 16
SM.APP.SESSION.CLEANUP.PRIORITY	Set request priority for cleanup App sessions request. **Allowed Values**: 0-31 **Default Value**: 20
SM.AUDIT.NOTIFY.PRIORITY	Set request priority for Audit notification request. **Allowed Values**: 0-31 **Default Value**: 30
SM.GET.APP.SESSION.PRIORITY	Set request priority for Retrieve App Session request. **Allowed Values**: 0-31 **Default Value**: 28
SM.GET.ASSOC.PRIORITY	Set request priority for retrieve Policy association request. **Allowed Values**: 0-31 **Default Value**: 28
SM.GET.SUBSCRIBER.SESSIONS.PRIORITY	Set request priority configured retrieve any type of session for rest api endpoint. **Allowed Values**: 0-31 **Default Value**: 28
SM.GET.ASSOC.QUERY.PRIORITY	Set request priority for advanced search of Policy Association request. **Allowed Values**: 0-31 **Default Value**: 28
CONGESTION_RESPONSE_CODE	Configure the response code of any rejected requests by SM service due to congestion state. By default, SM service sends 503 as response code. If configured then it shall be 5xx only. **Allowed Values**: 503 **Default Value**: 5xx

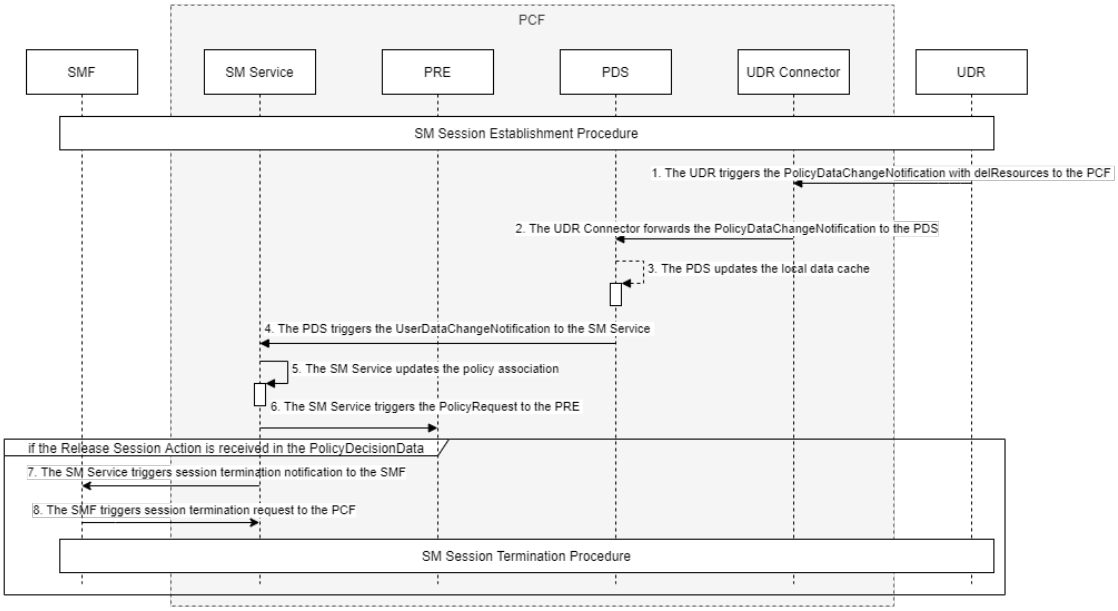
29. Click **Save** to save the PCF Session Management configurations.

7.3.2.1 UDR Subscriber Delete Resource Support

The PCF call flow has been updated to support terminating an active subscriber session by the deployed policy when the subscriber resources are deleted on the UDR.

The following figure describes a call flow for UDR subscriber resource deletion:

Figure 7-13 Call Flow for UDR Subscriber Resource Deletion

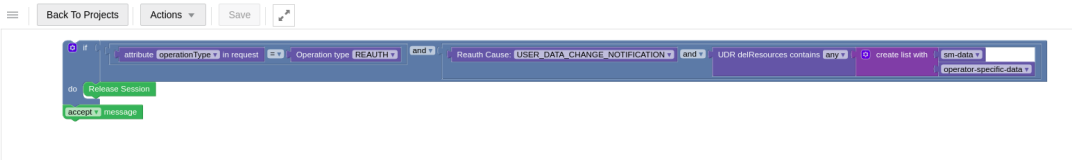


The call flow consists of the following tasks:

1. The UDR triggers the PolicyDataChangeNotification with the attribute "delResources" to the PCF.
2. The UDR connector forwards the received PolicyDataChangeNotification to the PDS.
3. After receiving the UDR notification, PDS calls UDR-Notify workflow, in this workflow, PDS compares the PolicyDataChangeNotification with the existing data in database and perform database update or delete, if needed, and notify the SM service (with the attribute "delResources"), if needed.
   If delResources exists in PolicyDataChangeNotification and the delResources URL meet the condition "contains '/policy-data/ues/{ueId}/operator-specific-data' or 'policy-data/ues/{ueId}/sm-data/' ", the existing subscriber information in database is be deleted.
4. After the data updates, the PDS triggers an UserDataChangeNotification with received PolicyDataChangeNotification to the SM service.
5. The related resources (smPolicyData or operatorSpecificData) within the SmPolicyAssociation are deleted when the delResources of the PolicyDataChangeNotification is received.
   The smPolicyData is deleted if the resource URL contains "sm-data".

   The operatorSpecificData is deleted if the resource URL contains "operator-specific-data".
6. The SM Service triggers the PolicyRequest with the UserDataChangeNotification to the PRE to get the PolicyDecisionData, if the "Release Session" action is received, Step 7 is triggered. New conditions and action have been added to support this.
   For more information on these conditions and action, see **PCF-SM Category** section in *Oracle Communcations Cloud Native Core, Converged Policy Design Guide*.

   The following screen capture illustrates an example of how to use these conditions and action to release a policy association:



7. The SM Service triggers session termination notification to the SMF if the "Release Session" action is received in the PolicyDecisionData.
8. The SMF triggers a session termination request to the PCF.

7.3.3 PCF Access and Mobility

This procedure provides information about configuring the Access and Mobility Service.

The **Access and Mobility** page displays the AM Service configurations. The page allows you to edit the configurations.

To configure the PCF Access and Mobility service:

1. From the navigation menu under **Policy**, navigate to **Service Configurations** and select **PCF Access and Mobility**.
   This opens the **PCF Access and Mobility** page. The page displays the existing configurations.
2. Click Edit .
   This opens the **PCF Access and Mobility** page.
3. Check the default configuration for the fields available in respective groups and edit as necessary.
   The following table describes the fields along with their valid input values under each group:

Table 7-31 Edit PCF Access and Mobility

Field Name	Description
System	
AMF Notification Retry Profile	Defines the retry profile configuration for Access and Mobility service. For more information about configuring retry profile, see Retry Profiles.
AMF Communication Profile	Specifies the NF communication profile used by AMF. For more information about configuring NF communication profiles, see NF Communication Profiles.
Enable Max Session Limit	Specifies whether to enable the Limiting the Number of Sessions functionality.
Max Session Limit Per User	Specifies the maximum number of sessions to be limited per user if **Enable Max Session Limit** flag is enabled. Range: 1-2 Default value: 1
Override Supported Features	This configuration enables the pending transactions along with the Advanced Settings.
User	

Table 7-31 (Cont.) Edit PCF Access and Mobility

Field Name	Description
Validate User	When this button is enabled , and the subscriber is not found in the UDR, or PCF is not able to query an available/eligible UDR, PCF sends an HTTP '400 USER_UNKNOWN' error to AMF. However, when you disable Validate User button, PCF continues policy processing even after the subscriber is not found in the UDR, or PCF is unable to query an available/eligible UDR. Default value: false.
Enable Subscriber State Variables	If this switch is enabled, AM Service controls the querying and saving of local and remote subscriber state variables (SSV) on PDS. Default value: false.
User Data Types	
AMPolicyData	If this switch is enabled, PCF fetches AMPolicyData from nUDR. For information on configuring AMPolicyData attributes, see **AMPolicyData Attributes table.**
CHF Data	If this switch is enabled, PCF fetches policy counters from CHF. For information on configuring CHF Data attributes, see **CHF Attributes table.**
AMPolicyData Attributes	
Subcribe to Notify	When **Subscribe to Notify** flag is enabled (default behavior), Policy subscribes with the UDR to receive notifications whenever a change is made to a subscriber profile. Further, AM service notifies PDS, which in turn looks for an existing subscription for that subsciber and do any of the following: • If a subscription already exists and `monitoredResourceUris` already contains "am-data", no action is taken by PDS. • If a subscription does not exist, PDS subscribes with the UDR using the POST method with monitoredResourceUris set to "am-data". By default, this option is disabled.
Query User on Create	When this parameter is enabled (default behavior), PDS queries the UDR about the subscriber contained in the AM Association create request. It is done by sending a GET request for "am-data" resource on the nudr-dr service. **Note:** If you enable the **Subscribe to Notify** parameter for AM service configurations, the Policy service may look up for the requested subscriber profile in its local cache before querying UDR. By default, this option is enabled.

Table 7-31 (Cont.) Edit PCF Access and Mobility

Field Name	Description
Query User on Update	When this parameter is enabled, PDS queries the UDR about the subscriber present in the AM Association update request by sending a GET request for “am-data” resource on the nudr-dr service. By default, this parameter is disabled. **Note**: If you enable the `Subscribe To Notify` flag for AM service configurations, the Policy service may look up for the requested subscriber profile in its local cache before querying UDR. When the `Subscribe To Notify` flag is enabled and previous subscription attempts failed, then along with the data query, PDS also sends a SUBSCRIBE request to UDR.
Query User on Terminate	When this parameter is enabled, PDS queries the UDR about the subscriber present in the AM Association delete request by sending a GET request for “am-data” resource on the nudr-dr service. **Note:** If you have enabled **Subscribe to Notify** button for AM service configurations, the PCF user service may look up for the requested subscriber profile in its local cache before querying UDR. By default, this button is disabled.
CHF Data Attributes	
CHF Data	If this switch is enabled, PCF fetches policy counters from CHF.
Async CHF Query Enabled	When this button is enabled, PCF interacts with CHF in Asynchronous mode.
Policy	
Evaluate	This determines if policy evaluate is enabled. By default, this button is enabled.
Rules	
Default Service Area Restriction	This field shows the default Service Area Restriction (SAR). To update your default Service Area Restriction, click **Edit** and select any one from the drop-down list of preconfigured SARs. For more information on how to create or configure SAR, see Service Area Restriction.
Default RFSP Index	The RFSP Index refers to a particular UE information used locally by the Access Network for implementing specific radio resource management strategies. This field shows the default value for RFSP Index, which can be a number between 1 and 256.

Table 7-31 (Cont.) Edit PCF Access and Mobility

Field Name	Description
Default Triggers	This field shows the default triggers. To update the default triggers, click **Edit** and select desired value(s) from drop-down list values. The supported values are: • LOC_CH, that is, change in location • PRA_CH, that is, change of UE presence in PRA
Audit	
Enabled	Determines whether to send registration request to Audit service or not. **Default Value**: True
Query AMF	When this flag is set to true, audit service sends notification to AMF to check the presence of a stale session. If a stale session is present, AMF sends a 404 response to audit service. Then, audit service deletes the AM Policy association from the database. **Default Value**: False **Note**: When Minimum Audit Attempts is 1 and Query AMF is false, Audit service removes the AM session after the maxTTL expiry and the Minimum Audit Attempts remains 0 in Audit database.
Notification Rate (per second)	Defines the number of stale records which Audit service notifies to AM service in one second. Value of this parameter ranges between 20 and 700. **Default and Recommended Value**: 50 **Note**: To configure higher number than the recommended value, contact My Oracle Support (https://support.oracle.com)
Policy Association Age (in minutes)	Defines the age of an AM policy association after which a record is considered to be stale on PCF and the AMF is queried for presence of such associations. Value of this parameter ranges between 1 and 10080. **Default Value**: 1440
Policy Association Maximum Age (in minutes)	Defines the maximum age of an AM policy association after which a record is purged from PCF AM d Value of this parameter ranges between 1 and 20160. **Default Value**: 2880

Table 7-31 (Cont.) Edit PCF Access and Mobility

Field Name	Description
Minimum Audit Attempts	Specifies the minimum number of consecutive failed audit attempts until maxTTL / forceTTL is reached. If maxTTL is reached and audit_attempts >= Minimum Audit Attempts for maxTTL, Audit service sends notification to AM service with maxTTL flag set to *true*. AM Service sends DELETE request to PDS. **Range**: 0-255 **Default Value**: 0 **Note**: If maxTTL is not reached and if audit attempts are reached, the number of audit attempts are incremented until maxTTL is reached.
Minimum Audit Passes Interval (in minutes)	Defines the time when next audit for the AM service table begins after delta time if auditing this table has been finished before this specified time. **Default Value**: 330
Pending Transaction	
UpdateNotify Retry Backoff (in milliseconds)	Defines the retry backoff for UpdateNotify in milliseconds. **Default Value**: 0 - 10000 milliseconds
UpdateNotify Retry Count	Defines the total number of retries for UpdateNotify. **Default Value**: 0 - 5

4. Perform the following steps to configure **Advanced Settings**:

   a. Click the ⊕ Add
   The page opens the **Add Advanced Settings** dialog box.

   b. In the dialog box, enter the following **keys** and respective **values**:

The following table describes the keys and values:

Table 7-32 Add Advanced Settings Configurations

Key	Value
CONCURRENCY.BULWARK_SERVICE_ENABLED	Enables the communication to Bulwark service, if it is set to true. **Default value**: False
CONCURRENCY.N15.UPDATE.LOCK_LEASE_DURATION	The duration for which lock is kept once the acquisition is successful. After this duration, the lock gets released automatically. **Default value**: 2000
CONCURRENCY.N15.UPDATE.LOCK_WAIT_DURATION	Duration by which the AM service waits for the response to get a lock. **Default value**: 0

Table 7-32 (Cont.) Add Advanced Settings Configurations

Key	Value
CONCURRENCY.N15.UPDATE.LOCK_REQUEST_RETRY_COUNT	Number of attempts the service make to request the lock. **Default value**: 2
CONCURRENCY.N15.UPDATE.LOCK_REQUEST_RETRY_BACKOFF	Defines the amount of time after which the service retries to gain the lock, incase of failure. **Default value**: 1000
CONCURRENCY.N15.UPDATE	Enables the communication to Bulwark service during UPDATE request flow. **Default value**: False
CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_LEASE_DURATION	The duration for which lock is kept once the acquisition is successful. After this duration, the lock gets released automatically. **Default value**: 2000
CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_WAIT_DURATION	Duration for which the Policy service will wait for the response. **Default value**: 0
CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_REQUEST_RETRY_COUNT	Number of attempts the service can make to request the lock. **Default value**: 2
CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_REQUEST_RETRY_BACKOFF	Number of retries to acquire the lock upon Failure response from Bulwark service for such a request. **Default value**: 1000
CONCURRENCY.N15.UPDATE_NOTIFY	Enables the communication to Bulwark service during update-notify request flow. **Default value**: False
BULWARK.SERVICE_CONNECTION_TIMEOUT	Allows to control the connection timeout of a Bulwark service. **Default value**: 3000
PENDING_TRANSACTION.NOTIFICATION_ERROR_HANDLING_APP	Allows to notify the pending transactions. **Default value**: 1
USER.RESET_CONTEXT_CHF_DATA_ON_AM_CREATE	Sets resetContext flag for the CHF on the request towards PDS. **Default value**: false
USER.RESET_CONTEXT_AM_POLICY_DATA_ON_AM_CREATE	Sets resetContext flag for the AM policy on the request towards PDS. **Default value**: false
CONCURRENCY.BULWARK_SERVICE_ENABLED	Enables the concurrency functionality. **Default value**: false
CONCURRENCY.N15.CREATE.ENABLED	Enables concurrency feature for AM-create call flow. **Default value**: false

Table 7-32 (Cont.) Add Advanced Settings Configurations

Key	Value
CONCURRENCY.N15.CREATE.LOCK_LEASE_DURATION	Lease duration for which lock will be acquired. After this time, Bulwark will automatically release the lock. **Default value**: 2000
CONCURRENCY.N15.CREATE.LOCK_WAIT_DURATION	Wait time out for lock acquisition if not acquired. **Default value**: 0
CONCURRENCY.N15.CREATE.LOCK_REQUEST_RETRY_COUNT	Wait time out for lock acquisition if not acquired. **Default value**: 2
CONCURRENCY.N15.CREATE.LOCK_REQUEST_RETRY_BACKOFF	This is the duration for which AM service will wait once it receives ALREADY_LOCKED Response from Bulwark for the AM Update flow. **Default value**: 1000
CONCURRENCY.N15.TERMINATE.ENABLED	Enables concurrency feature for AM-Delete call flow. **Default value**: false
CONCURRENCY.N15.TERMINATE.LOCK_LEASE_DURATION	Lease duration for which lock will be acquired. After this time, Bulwark will automatically release the lock. **Default value**: 2000
CONCURRENCY.N15.TERMINATE.LOCK_REQUEST_RETRY_BACKOFF	This is the duration for which AM service will wait once it receives ALREADY_LOCKED Response from Bulwark for the AM Delete flow. **Default value**: 1000
CONCURRENCY.N15.TERMINATE.LOCK_REQUEST_RETRY_COUNT	If lock acquisition failed from Bulwark due to lock request failed. **Default value**: 2
CONCURRENCY.N15.TERMINATE.LOCK_WAIT_DURATION	Wait time out for lock acquisition if not acquired. **Default value**: 0
CONCURRENCY.N15.UPDATE.ENABLED	Enable concurrency feature for AM-Update call flow. **Default value**: false
CONCURRENCY.N15.UPDATE.LOCK_LEASE_DURATION	Lease duration for which lock will be acquired. After this time, Bulwark will automatically release the lock. **Default value**: 2000
CONCURRENCY.N15.UPDATE.LOCK_REQUEST_RETRY_BACKOFF	This is the duration for which AM service will wait once it receives ALREADY_LOCKED Response from Bulwark for the AM Update flow. **Default value**: 1000
CONCURRENCY.N15.UPDATE.LOCK_REQUEST_RETRY_COUNT	If lock acquisition failed from Bulwark due to lock request failed. **Default value**: 2
CONCURRENCY.N15.UPDATE.LOCK_WAIT_DURATION	Wait time out for lock acquisition if not acquired. **Default value**: 0

Table 7-32 (Cont.) Add Advanced Settings Configurations

Key	Value
CONCURRENCY.N15.UPDATE_NOTIFY.ENABLED	Enable concurrency feature for AM-Update-Notify call flow. **Default value**: false
CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_LEASE_DURATION	Lease duration for which lock will be acquired. After this time, Bulwark will automatically release the lock. **Default value**: 2000
CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_REQUEST_RETRY_BACKOFF	This is the duration for which AM service will wait once it receives ALREADY_LOCKED Response from Bulwark for the AM Update Notify flow. **Default value**: 1000
CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_REQUEST_RETRY_COUNT	If lock acquisition failed from Bulwark due to lock request failed. **Default value**: 2
CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_WAIT_DURATION	Wait time out for lock acquisition if not acquired. **Default value**: 0
RESOURCEID.SUFFIXLIST	ResourceId because this is generic to all resources like AM or other service..Like in our case for AM we need this for association id which is a resource identifier.Hence ResourceId. Suffix because we are going to concatenate it with association id which will be after generated uuid. **Default value**: SUPI

Table 7-32 (Cont.) Add Advanced Settings Configurations

Key	Value
ENABLE.ASSOCIATIONID.ENCODING	Enables the encoding of the Association ID in the format: `PolicyAssociationId_versionNo_encoded(featureName1:value1)_encoded(featureName2:value2)` where: **versionNo**: is the value which will decide about the encoding version being used. If the value is 0, it indicates that it will be base64 encrypted. **encoded**: indicates the encoded value for each feature, hence each feature will be separated by an underscore, that is "_". An encoded value consists of a number that will mean the name of the feature, in this case 0 will be for Concurrency and value separated by ":" **featureName**: represents the feature for which the coding is, in the format it will be represented as a number that will have a feature assigned to it. For example, **value**: is key value pair of ***SUPI/GPSI*** **Default value**: false
CONCURRENCY.N15.CREATE.ALLOW_ON_SERVICE_FAILURE	If the flag is set to true, the AM Create request is allowed to be processed further even after all the lock acquisition retries with Bulwark Service end up with an error. If this flag is set to false, the AM Create request is responded with 500 Internal Service Error, when all the lock acquisition retries with Bulwark Service end up with an error. **Default value**: true
CONCURRENCY.N15.UPDATE.ALLOW_ON_SERVICE_FAILURE	If the flag is set to true, the AM Update request is allowed to be processed further even after all the lock acquisition retries with Bulwark Service end up with an error. If this flag is set to false, the AM Update request is responded with 500 Internal Service Error, when all the lock acquisition retries with Bulwark Service end up with an error. **Default value**: true

Table 7-32 (Cont.) Add Advanced Settings Configurations

Key	Value
CONCURRENCY.N15.UPDATE_NOTIFY.ALLOW_ON_SERVICE_FAILURE	If the flag is set to true, the AM Update Notify and Update Terminate requests are allowed to be processed further even after all the lock acquisition retries with Bulwark Service end up with an error. If this flag is set to false, the AM Update Notify and Update Terminate requests are responded with 500 Internal Service Error, when all the lock acquisition retries with Bulwark Service end up with an error. **Default value**: true
CONCURRENCY.N15.TERMINATE.ALLOW_ON_SERVICE_FAILURE	If the flag is set to true, the AM Terminate request is allowed to be processed further even after all the lock acquisition retries with Bulwark Service end up with an error. If this flag is set to false, the AM Terminate request is responded with 500 Internal Service Error, when all the lock acquisition retries with Bulwark Service end up with an error. **Default value**: true
CONCURRENCY.N15.CLEANUP.ENABLED	Enables concurrency feature for AM cleanup call flow. **Default value**: false
CONCURRENCY.N15.CLEANUP.LOCK_REQUEST_RETRY_COUNT	If lock acquisition failed from Bulwark due to lock request failed. **Default value**: 2
CONCURRENCY.N15.CLEANUP.LOCK_REQUEST_RETRY_BACKOFF	This is the duration for which AM service will wait once it receives ALREADY_LOCKED Response from Bulwark for the AM Cleanup flow. **Default value**: 1000
CONCURRENCY.N15.CLEANUP.LOCK_LEASE_DURATION	Lease duration for which lock will be acquired. After this time, Bulwark will automatically release the lock. **Default value**: 2000
CONCURRENCY.N15.CLEANUP.LOCK_WAIT_DURATION	Wait timeout for lock acquisition if not acquired. **Default value**: 3000
CONCURRENCY.N15.CLEANUP.ALLOW_ON_SERVICE_FAILURE	If bulwark lock is not obtained after all retries for the AM Cleanup flow, then if enabled, the request will continue. If disabled, the request will be rejected. **Default value**: true

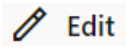
5. Click **Save**.
   The page saves the PCF Access and Mobility configurations.

7.3.4 PCF Policy Authorization

This procedure provides information about configuring the PCF Policy Authorization Service.

The **PCF Policy Authorization** page displays the authorization service configurations. The page allows you to edit the configurations.

To configure the PCF Policy Authorization Service:

1. From the navigation menu under **Policy**, navigate to **Service Configurations**, and select **PCF Policy Authorization**.
   This opens the **PCF Policy Authorization** page. The page displays the existing configurations.
2. Click 

   .
   This opens the **Edit PCF Policy Authorization** page.
3. Check the default configuration for all the fields in all groups and edit as necessary.
   The following table describes the input fields displayed under each group:

Table 7-33 Edit PCF Policy Authorization

Field Name	Description
System	
Af Direct Reply	Determines, if reply must be sent to AF before the PCF that sends the policy decision to SMF, after successful session binding. **Default Value**: `true`
Override Supported Features	Defines the supported features that can be configured to override system embedded values. Represent a string separated by comma. For example, "InfluenceOnTrafficRouting,SponsoredConnectivity". The "" means an empty supported feature set.
AF Terminate Uri Segment	Specifies AF notify to terminate uri segment appended to AF notify uri. For example, {NotificationUri}/termination. "termination" is the terminate uri segment.**Default Value**: `terminate`
AF Subscriber Notify Segment	Specifies AF subscription notify uri segment appended to AF subs notify uri. For example, {NotificationUri}/notify. "notify" is the subscription notify uri segment.**Default Value**: `terminate`

Table 7-33 (Cont.) Edit PCF Policy Authorization

Field Name	Description
Rx Resource Allocation Partial Failure Report Prefence	After PCF triggers a notification to Diameter Connector, the connector generates a RAR message. The partial failed specific action in the RAR message depends on the priority of the action subscribed in the AAR message. The priority of the actions is INDICATION_OF_FAILED_RESOURCES_ALLOCATION > INDICATION_OF_RELEASE_OF_BEARER > INDICATION_OF_LOSS_OF_BEARER. If you want to assign the action not depend on the priority, you can assign the action in this field. The default configuration is empty, you can choose one action from the following options. Once the value is defined in this field, the connector uses the configured action not depend on the priority. Valid Options are: • INDICATION_OF_FAILED_RESOURCES_ALLOCATION • INDICATION_OF_RELEASE_OF_BEARER • INDICATION_OF_LOSS_OF_BEARER
AF Notifications Retry Profile	Defines the retry profile configuration for Policy Authorization. For more information on configuring retry profile, see Retry Profiles.
AF Communication Profile	Specifies the NF communication profile used by AF. For more information about configuring NF communication profiles, see NF Communication Profiles.
Data Compression Scheme	
Enable Authorization Lifetime	Specifies whether to enable feature negotiation for Authorization Lifetime of Rx Sessions. **Default Value**: false
IMS Emergency Session	
Emergency Service URNs	Defines the Uniform Resource Name (URN) values for emergency and other well-known services.
Reservation Priority Types	Detemines the IMS Signalling Priority.**Default Value**: PRIO_6
NWDAF	
Enable NWDAF Data	Enables or disables the **Enable NWDAF Data** switch.
Audit	
Audit Enabled	Specifies whether to enable or disable the stale Rx session audit and deletion. When this parameters is enabled, SM Service registers with Audit service for monitoring Rx sessions. **Default Value**: false
App Session Age (in minutes)	Specifies the Time To Live (TTL) of Rx sessions. **Range**: 0-21600 **Default Value**: 1440

Table 7-33 (Cont.) Edit PCF Policy Authorization

Field Name	Description
App Session Maximum Age (in minutes)	Specifies the maximum Time To Live (TTL) of Rx sessions. **Default Value**: 4320 **Range**: 0-21600
Notification Rate (per second)	Maximum amount of app session cleanup requests per second. **Default and Recommended Value**: 50 **Note**: To configure higher number than the recommended value, contact My Oracle Support (https://support.oracle.com)
Minimum Audit Passes Interval (in minutes)	Specifies how long the Audit Service waits for next audit cycle. **Default Value**: 10
Handle NULL As Stale	If `EXPIRY_TIMESTAMP` or `SITEID` contains `NULL` value, (legacy stale records after upgrade) they will be considered in the next audit cycle as TTL expired. **Default Value**: false

4. Perform the following steps to configure **Advanced Settings**:
   a. Click the ⊕ Add
      The page opens the **Add Advanced Settings** dialog box.
   b. In the dialog box, enter the following **keys** and respective **values**:

The following table describes the keys and values:

Table 7-34 Add Advanced Settings Configurations

Key	Value
PA.NWDAF.EVENT	Allow to add one or more Events to be send in the request towards NWDAF. **Default value**: SLICE_LOAD_LEVEL
PA.NWDAF.REQUEST_TIMEOUT	Allow to control the timeout of the request from SM towards NWDAF. **Default value**: 3000 (ms)
SYSTEM.PA_ERROR_HANDLER_ENABLED	Allows to enable the RAA error handling feature. **Default value**: False
SYSTEM.PA.EVENT_TRIGGERS	This is a comma separated list of Event-Trigger names that are installed on SMF by PCF upon PA session establishment. **Default value**: Empty ("")
SYSTEM.RX.EVENT_TRIGGERS	This is a comma separated list of Event-Trigger names that are installed on SMF by PCF upon Rx session establishment. **Default value**: ACCESS_TYPE_CHANGE

5. Click **Save**.
   The page saves the PCF Policy Authorization configurations.

7.3.5 PCF UE Policy Service

This procedure provides information about configuring the PCF UE Policy Service.

The **PCF UE Policy** page displays the UE Service configurations. The page allows you to edit the configurations.

To configure PCF UE Policy service:

1. From the navigation menu under **Policy** navigate to **Service Configurations** and select **PCF UE Policy**.
   This opens the **PCF UE Policy** page. The page displays the existing configurations.
2. Click 

   .
   This opens the **Edit PCF UE Policy** page.
3. Check the default configuration for the fields available in the respective groups and edit as necessary.
   The following table describes each field present under PCF UE policy along with the default configurations.

Table 7-35 Edit PCF UE Policy

Field Name	Description
System	
Notification URI Root	Specifies the URI Root where the PCF (UE Policy service) is expected to receive notifications from AMF. **Example**: `http://localhost:8080` **Note:** Include the API prefix required by the API Gateway (if any). For example, `https://api-gateway.pcf.com/uepolicyservice`
Enable Max Session Limit	Specifies whether to enable the Limiting the Number of Sessions functionality.
Max Session Limit Per User	Specifies the maximum number of sessions to be limited per user if **Enable Max Session Limit** flag is enabled. Range: 1-2 Default value: 1
Override Supported Features	This configuration enables the pending transactions along with the Advanced Settings.
AMF	
AMF Discovery Criteria	Specifies the criteria to discover producer AMFs based on the AMF Set ID and the AMF Region ID. Valid options are:GUAMISetID and RegionID. When this parameter is set to GUAMI, Policy parses the GUAMI to fetch the AMF SetID and AMF RegionID and uses these IDs to discover the producer AMF.
AMF Communication Profile	Specifies the NF communication profile used by AMF. For more information about configuring NF communication profiles, see NF Communication Profiles.
AMF Notification Retry Profile	Specifies the retry profile used by AMF. For more information about configuring retry profiles, see Retry Profiles.

Table 7-35 (Cont.) Edit PCF UE Policy

Field Name	Description
NAS Communication Profile	Specifies the NF communication profile used by NAS. For more information about configuring NF communication profiles, see NF Communication Profiles.
NAS Retry Profile for Initial Messages	Specifies the retry profile used by NAS. For more information about configuring NF retry profiles, see Retry Profiles.
NAS Retry Profile for Subsequent Messages	Specifies the retry profile for subsequent messages used by NAS. For more information about configuring NF retry profiles, see Retry Profiles.
Send SBI Binding Header For N1N2 Transfer Requests	Indicates when to send the SBI binding information for N1N2Transfer request. The options can be: • When binding information changes • Always • Never **Default Value**: When binding information changes
Send Routing Binding Header For N1N2 Transfer Requests	Indicates when to send the routing binding information for N1N2Transfer request. The options can be: • Always • Never **Default (no value)**
Send Discovery Header For N1N2 Transfer Requests	Indicates when to send the discovery binding information for N1N2Transfer request. The options can be: • Always • Never **Default (no value)**
User	
Validate User	When **Validate User** is enabled, and the subscriber is not found in the UDR, or PCF is not able to query an available/eligible UDR, PCF fails the UE Association creation request with a 400 USER_UNKNOWN error. On the contrary when you disable **Validate User**, and the subscriber is not found in the UDR, or PCF is not able to query an available/eligible UDR, PCF does not fail the UE Association creation request, but continue policy processing. **Default Value**: `disabled`
Enable Subscriber State Variables	If this switch is enabled, UE Policy Service controls the querying and saving of local and remote subscriber state variables (SSV) on PDS. **Default value**: false
User Data Types	
UEPolicyData	If this switch is enabled, PCF fetches UEPolicyData from nUDR. For information on configuring UEPolicyData attributes, see **UEPolicyData Attributes table.**
CHF Data	If this switch is enabled, PCF fetches policy counters from CHF. For information on configuring CHF Data attributes, see **CHF Attributes table.**
UEPolicyData Attributes	

Table 7-35 (Cont.) Edit PCF UE Policy

Field Name	Description
Subscribe to Notify	When **Subscribe to Notify** is enabled, Policy subscribes with the UDR to get notified on changes in subscriber profile. UE Policy notifies PDS, which in turn looks for an existing subscription for that subscriber and do any of the following: • If a subscription already exists and the `monitoredResourceUris` already contains "ue-policy-set", no action is taken by PDS. • If a subscription does not exist, PDS subscribes with the UDR using the POST method with monitoredResourceUris set to "ue-policy-set". **Default Value**: The button is `enabled` by default.
Query User on Create	Determines if user query from UDR is enabled. When this option is enabled, PDS queries the UDR about the subscriber contained in the UE Association create request by sending a GET request for “ue-policy-set” resource on the nudr-dr service. **Note:** The Policy Service caches the subscriber profile when **Subscribe To Notify** option is enabled. In that case, Policy may not always reach the UDR when the subscriber profile is found in the local cache. **Default Value**: The parameter is `enabled` by default.
Query User on Update	Determines if user query from UDR on update is enabled. When this option is enabled, PDS queries the UDR about the subscriber present in the UE Association update request by sending a GET request for “ue-policy-set” resource on the nudr-dr service. **Default Value**: This parameter is `disabled` by default. **Note**: The Policy Service caches the subscriber profile when the `Subscribe To Notify` option is enabled. In that case, Policy may not always reach the UDR when the subscriber profile is found in the local cache. When the `Subscribe To Notify` flag is enabled and previous subscription attempts failed, then along with the data query, PDS also sends a SUBSCRIBE request to UDR.
Query User on Terminate	Determines if user query from UDR on delete is enabled. When this option is enabled, PCF queries the UDR about the subscriber present in the UE Association delete request by sending a GET request for “ue-policy-set” resource on the nudr-dr service. **Note:** The PCF User Service caches the subscriber profile when the **Subscribe To Notify** option is enabled, in that case, the PCF may not always reach the UDR when the subscriber profile is found in the local cache. **Default Value**: The button is `disabled` by default.

Table 7-35 (Cont.) Edit PCF UE Policy

Field Name	Description
CHF Data Attributes	
CHF Data	If this switch is enabled, PCF fetches policy counters from CHF.
Async CHF Query Enabled	When this button is enabled, PCF interacts with CHF in Asynchronous mode.
Home PLMN	
MCC	The Mobile Country Code and Mobile Network Code of the Home PLMN **Default Value**: NA
MNC	The Mobile Country Code and Mobile Network Code of the Home PLMN **Default Value**: NA
N1 Message Transfer Settings	
N1 Message Maximum Size	Maximum number of URSP Rules to be encoded in a single UE Policy Section (UPSI) container. **Default Value**: 2000
UE Policy Section Maximum Size	The maximum size in bytes of a "MANAGE UE POLICY COMMAND" after encoding to send out to AMF for delivery to UE. **Default Value**: 2000
Maximum Number of URSP Rules per UE Policy Section	Maximum number of URSP Rules to be encoded in a single UE Policy Section (UPSI) container. **Default Value**: 4
UE Policy Section Code (UPSC) Start	This fields indicate the range of UPSC allocation by PCF and can be used to maintain a dedicated range of UPSCs used by PCF generated UPSIs. Configuring these fields avoids the conflict of UPSCs between the UE built-in UPSC and the PCF delivered UPSC. **Note:** If the UPSI based policy action is also used for non-fragmented URSP delivery, then the range of UPSCs used by that method should be outside the range configured here. **Default Value**: Start: 0

Table 7-35 (Cont.) Edit PCF UE Policy

Field Name	Description
UE Policy Section Code (UPSC) End	This fields indicate the range of UPSC allocation by PCF and can be used to maintain a dedicated range of UPSCs used by PCF generated UPSIs. Configuring these fields avoids the conflict of UPSCs between the UE built-in UPSC and the PCF delivered UPSC. **Note:** If the UPSI based policy action is also used for non-fragmented URSP delivery, then the range of UPSCs used by that method should be outside the range configured here. **Default Value**: End: 65535
Timer Settings	
T 3501 Timer Duration	If it's value is zero, T 3501 Timer handling functionality will be disabled. If a non-zero value is configured, then the timer will be initiated after every N1 transfer receives success response from AMF. PCF will be expecting N1N2-notify to be received before the timer gets expired. **Default Value**: 5000 **Range**: 0 to 60000
Back-off Timer Duration	The action to be taken by PCF when the AMF notifies the PCF that it could not deliver the "MANAGE UE POLICY COMMAND" message to UE. **Default Value**: 15000
N1 Message Retransmission Settings	
On T 3501 Timer Expiry	
Action	The action to be taken by PCF when N1 notification is not received within configured duration defined in T 3501 Timer Duration. **Default Value**: Abort N1 Delivery
Max Number of Re-transmissions	Maximum number of retransmissions to be attempted (excluding the initial attempt) after which UE service shall not try further to resend the same fragment to AMF. **Default Value**: 2
Re-transmission Failure Behaviour	The behavior of PCF UE service if the N1 message is not delivered to the UE after a maximum number of re-transmissions have been tried. **Default Value**: Abort N1 Delivery
On Transaction Failure Notification	
Action	The action to be taken by PCF when the AMF notifies the PCF that it could not deliver the "MANAGE UE POLICY COMMAND" message to UE. **Default Value**: Abort N1 Delivery

Table 7-35 (Cont.) Edit PCF UE Policy

Field Name	Description
Max Number of Re-transmissions	Maximum number of re-transmissions to be attempted (excluding the initial attempt) after which UE service shall not try further to re-send the same fragment to AMF. **Default Value**: 2
Re-transmission Failure Behaviour	The behavior of PCF UE service if the N1 message is not delivered to the UE after a maximum number of re-transmissions have been tried. **Default Value**: Abort N1 Delivery
On UE Policy Command Reject	
Action	The action to be taken by PCF when the UE notifies the PCF that it could not process the "MANAGE UE POLICY COMMAND" message. **Default Value**: Abort N1 Delivery
Maximum Number of Re-transmissions	Maximum number of re-transmissions to be attempted (excluding the initial attempt) after which UE service shall not try further to re-send the same fragment to AMF **Default Value**: 2
Re-transmission Failure Behaviour	The behavior of PCF UE service if the N1 message is not delivered to the UE after a maximum number of re-transmissions have been tried. **Default Value**: Abort N1 Delivery
Audit	
Enabled	If this flag is enabled, UE service registers with Audit service for auditing the records in `UEPolicyAssociation` table. Once the registration is successful, a record for UE service is created in the `AuditRegistration` table. **Default Value**: False
Query AMF	If this flag is enabled, the UE service sends the stale session notification received from Audit service to AMF to check if the session is present in AMF. If a stale session is present, AMF sends a 404 response to audit service. Then, audit service deletes the UE Policy association from the database. **Default Value**: False **Note**: When Minimum Audit Attempts is 1 and Query AMF is false, Audit service removes the UE Policy session after the maxTTL expiry and the Minimum Audit Attempts remains 0 in Audit database.
Notification Rate (per second)	Defines the maximum number of stale records, which Audit Service notifies to UE service in one second. Value of this parameter ranges between 20 and 700. **Default and Recommended Value**: 50 **Note**: To configure higher number than the recommended value, contact My Oracle Support (https://support.oracle.com)

Table 7-35 (Cont.) Edit PCF UE Policy

Field Name	Description
Policy Association Age (in minutes)	Defines the age of the UE Policy Association, after which a record is considered to be stale on PCF and the AMF is queried (if query AMF is set to TRUE) for presence of such associations. Value of this parameter ranges between 1 and 10080. **Default Value**: 1440
Policy Association Maximum Age (in minutes)	Defines the maximum age of a UE Policy Association, after which a record is considered as stale and is purged from PCF UE database without sending further queries to AMF. Value of this parameter ranges between 1 and 20160. **Default Value**: 2880
Minimum Audit Attempts	Specifies the minimum number of consecutive failed audit attempts until maxTTL / forceTTL is reached. If maxTTL is reached and audit_attempts >= Minimum Audit Attempts for maxTTL, Audit service sends notification to UE Policy service with maxTTL flag set to *true*. UE Policy service sends DELETE request to PDS. **Range**: 0-255 **Default Value**: 0 **Note**: If maxTTL is not reached and if audit attempts are reached, the number of audit attempts are incremented until maxTTL is reached.
Minimum Audit Passes Interval (in minutes)	Defines the time when next audit for the UE service table is done after delta time if auditing this table has been finished before this specified time. Value of this parameter ranges between 1 and 1440. **Default Value**: 10
Pending Transaction	
UpdateNotify Retry Backoff (in milliseconds)	Defines the retry backoff for UpdateNotify in milliseconds. **Default Value**:
UpdateNotify Retry Count	Defines the total number of retries for UpdateNotify. **Default Value**:

4. Perform the following steps to configure **Advanced Settings**:

   a. Click the ⊕ Add
   The page opens the **Add Advanced Settings** dialog box.

   b. In the dialog box, enter the following **keys** and respective **values**:

The following table describes the keys and values:

Table 7-36 Add Advanced Settings Configurations

Key	Value
CONCURRENCY.ENABLED	Enables the communication to Bulwark service, if it is set to true. These will override Bulwark service enabled flag set during deployment using deployment yaml files. **Default value**: False

Table 7-36 (Cont.) Add Advanced Settings Configurations

Key	Value
CONCURRENCY.N15.CREATE.LOCK_LEASE_DURATION	The duration for which lock is kept once the acquisition is successful. After this duration, the lock gets released automatically. **Default value**: 2000
CONCURRENCY.N15.CREATE.LOCK_WAIT_DURATION	Duration by which the UE service waits for the response to get a lock. **Default value**: 0
CONCURRENCY.N15.CREATE.LOCK_REQUEST_RETRY_COUNT	Number of attempts the service make to request the lock. **Default value**: 2
CONCURRENCY.N15.CREATE.ENABLED	Enables the communication to Bulwark service during CREATE request flow if these flag and Bulwark service flag in deployment yaml file are set to True. **Default value**: False
CONCURRENCY.N15.CREATE.ALLOW_ON_SERVICE_FAILURE	Allows the UE service to continue with the processing of CREATE request even if the response get from Bulwark service to lock/unlock was a failure. **Default value**: True
CONCURRENCY.N15.UPDATE.LOCK_LEASE_DURATION	Duration of the lock is kept once the block is successful. After this duration the lock will be released automatically. The duration is set only for UPDATE request flow. It can be set to 0 to indicate no release lock timeout. **Default value**: 2000
CONCURRENCY.N15.UPDATE.LOCK_WAIT_DURATION	Duration which consumer service will wait for the lock response. The duration is set only for UPDATE request flow. It can be set to 0 to indicate no release lock timeout. **Default value**: 0
CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_REQUEST_RETRY_COUNT	Number of retries when a request to bulwark service fails. The duration is set only for UPDATE request flow. **Default value**: 2
CONCURRENCY.N15.UPDATE.ENABLED	Enables the communication to Bulwark service during UPDATE request flow. IF pendingtransaction feature is enabled then this Advance setting is avoided. **Default value**: False
CONCURRENCY.N15.UPDATE.ALLOW_ON_SERVICE_FAILURE	Allows the UE service to continue with the processing of UPDATE request even if the response get from Bulwark service to lock/unlock was a failure. **Default value**: True
CONCURRENCY.N15.TERMINATE.LOCK_LEASE_DURATION	Duration of the lock is kept once the block is successful. After this duration the lock will be released automatically. The duration is set only for TERMINATE request flow. It can be set to 0 to indicate no release lock timeout. **Default value**: 2000
CONCURRENCY.N15.TERMINATE.LOCK_WAIT_DURATION	Duration of the lock is kept once the block is successful. After this duration the lock will be released automatically. The duration is set only for TERMINATE request flow. It can be set to 0 to indicate no release lock timeout. **Default value**: 0
CONCURRENCY.N15.TERMINATE.LOCK_REQUEST_RETRY_COUNT	Number of retries when a request to bulwark service fails. The duration is set only for TERMIANATE request flow. **Default value**: 2

Table 7-36 (Cont.) Add Advanced Settings Configurations

Key	Value
CONCURRENCY.N15.TERMINATE.ENABLED	Enables the communication to Bulwark service during TERMINATE request flow if theseflag and Bulwark service flag in deployment yaml file are set to True. **Default value**: False
CONCURRENCY.N15.TERMINATE.ALLOW_ON_SERVICE_FAILURE	Allows the UE service to continue with the processing of TERMINATE request even if the response get from Bulwark service to lock/unlock was a failure **Default value**: True
CONCURRENCY.N15.NOTIFICATION.LOCK_LEASE_DURATION	Duration of the lock is kept once the block is successful. After this duration the lock will be released automatically. The duration is set only for NOTIFY request flow. It can be set to 0 to indicate no release lock timeout. **Default value**: 2000
CONCURRENCY.N15.NOTIFICATION.LOCK_WAIT_DURATION	Duration which consumer service will wait for the lock response. The duration is set only for NOTIFY request flow. It can be set to 0 to indicate no release lock timeout. **Default value**: 0
CONCURRENCY.N15.NOTIFICATION.LOCK_WAIT_DURATION	Duration which consumer service will wait for the lock response. The duration is set only for NOTIFY request flow. It can be set to 0 to indicate no release lock timeout. **Default value**: 2
CONCURRENCY.N15.NOTIFICATION.ENABLED	Set it to True will enable the communication to Bulwark service during NOTIFY request flow. IF pendingtransaction feature is enabled then this Advance setting is avoided. **Default value**: False
CONCURRENCY.N15.NOTIFICATION.ALLOW_ON_SERVICE_FAILURE	Allows the UE service to continue with the processing of NOTIFY request even if the GET response from Bulwark service to lock/unlock was a failure. **Default value**: True
BULWARK.SERVICE_CONNECTION_TIMEOUT	Duration which consumer service will wait for Bulwark response. **Default value**: 3000
USER.RESET_CONTEXT_SSV_ON_UE_CREATE	Sets resetContext flag for the SSV on the request towards PDS. **Default value**: false
USER.RESET_CONTEXT_CHF_DATA_ON_UE_CREATE	Sets resetContext flag for the CHF on the request towards PDS. **Default value**: false
USER.RESET_CONTEXT_UE_POLICY_DATA_ON_UE_CREATE	Sets resetContext flag for the UE policy on the request towards PDS. **Default value**: false
CONCURRENCY.BULWARK_SERVICE_ENABLED	Enables the concurrency functionality. **Default value**: false
CONCURRENCY.N15.CREATE.ENABLED	Enables concurrency feature for AM-create call flow. **Default value**: false
CONCURRENCY.N15.CREATE.LOCK_LEASE_DURATION	Lease duration for which lock will be acquired. After this time, Bulwark will automatically release the lock. **Default value**: 2000
CONCURRENCY.N15.CREATE.LOCK_WAIT_DURATION	Wait time out for lock acquisition if not acquired. **Default value**: 0

Table 7-36 (Cont.) Add Advanced Settings Configurations

Key	Value
CONCURRENCY.N15.CREATE.LOCK_REQUEST_RETRY_COUNT	Wait time out for lock acquisition if not acquired. **Default value**: 2
CONCURRENCY.N15.CREATE.LOCK_REQUEST_RETRY_BACKOFF	This is the duration for which UEservice will wait once it receives ALREADY_LOCKED Response from Bulwark for the UE Update flow. **Default value**: 1000
CONCURRENCY.N15.TERMINATE.ENABLED	Enables concurrency feature for AM-Delete call flow. **Default value**: false
CONCURRENCY.N15.TERMINATE.LOCK_LEASE_DURATION	Lease duration for which lock will be acquired. After this time, Bulwark will automatically release the lock. **Default value**: 2000
CONCURRENCY.N15.TERMINATE.LOCK_REQUEST_RETRY_BACKOFF	This is the duration for which UE service will wait once it receives ALREADY_LOCKED Response from Bulwark for the UE Delete flow. **Default value**: 1000
CONCURRENCY.N15.TERMINATE.LOCK_REQUEST_RETRY_COUNT	If lock acquisition failed from Bulwark due to lock request failed. **Default value**: 2
CONCURRENCY.N15.TERMINATE.LOCK_WAIT_DURATION	Wait time out for lock acquisition if not acquired. **Default value**: 0
CONCURRENCY.N15.UPDATE.ENABLED	Enable concurrency feature for AM-Update call flow. **Default value**: false
CONCURRENCY.N15.UPDATE.LOCK_LEASE_DURATION	Lease duration for which lock will be acquired. After this time, Bulwark will automatically release the lock. **Default value**: 2000
CONCURRENCY.N15.UPDATE.LOCK_REQUEST_RETRY_BACKOFF	This is the duration for which UE service will wait once it receives ALREADY_LOCKED Response from Bulwark for the UE Update flow. **Default value**: 1000
CONCURRENCY.N15.UPDATE.LOCK_REQUEST_RETRY_COUNT	If lock acquisition failed from Bulwark due to lock request failed. **Default value**: 2
CONCURRENCY.N15.UPDATE.LOCK_WAIT_DURATION	Wait time out for lock acquisition if not acquired. **Default value**: 0
CONCURRENCY.N15.UPDATE_NOTIFY.ENABLED	Enable concurrency feature for AM-Update-Notify call flow. **Default value**: false
CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_LEASE_DURATION	Lease duration for which lock will be acquired. After this time, Bulwark will automatically release the lock. **Default value**: 2000
CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_REQUEST_RETRY_BACKOFF	This is the duration for which UE service will wait once it receives ALREADY_LOCKED Response from Bulwark for the UE Update Notify flow. **Default value**: 1000
CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_REQUEST_RETRY_COUNT	If lock acquisition failed from Bulwark due to lock request failed. **Default value**: 2
CONCURRENCY.N15.UPDATE_NOTIFY.LOCK_WAIT_DURATION	Wait time out for lock acquisition if not acquired. **Default value**: 0

Table 7-36 (Cont.) Add Advanced Settings Configurations

Key	Value
RESOURCEID.SUFFIXLIST	ResourceId because this is generic to all resources like UE or other service. Like in our case for UE we need this for association id which is a resource identifier.Hence ResourceId. Suffix because we are going to concatenate it with association id which will be after generated uuid. **Default value**: SUPI
ENABLE.ASSOCIATIONID.ENCODING	Enables the encoding of the Association ID in the format: `PolicyAssociationId_versionNo_encoded(featureName1:value1)_encoded(featureName2:value2)` where: **versionNo**: is the value which will decide about the encoding version being used. If the value is 0, it indicates that it will be base64 encrypted. **encoded**: indicates the encoded value for each feature, hence each feature will be separated by an underscore, that is "_". An encoded value consists of a number that will mean the name of the feature, in this case 0 will be for Concurrency and value separated by ":" **featureName**: represents the feature for which the coding is, in the format it will be represented as a number that will have a feature assigned to it. For example, **value**: is key value pair of ***SUPI/GPSI*** **Default value**: false
CONCURRENCY.N15.CREATE.ALLOW_ON_SERVICE_FAILURE	If this flag is set to true, the UE Create request is allowed to be processed further even after all the lock acquisition retries with Bulwark Service end up with an error. If this flag is set to false, the UE Create request is responded with 500 Internal Service Error, when all the lock acquisition retries with Bulwark Service end up with an error. **Default value**: true
CONCURRENCY.N15.UPDATE.ALLOW_ON_SERVICE_FAILURE	If this flag is set to true, the UE Update request is allowed to be processed further even after all the lock acquisition retries with Bulwark Service end up with an error. If this flag is set to false, the UE Update request is responded with 500 Internal Service Error, when all the lock acquisition retries with Bulwark Service end up with an error. **Default value**: true
CONCURRENCY.N15.T3501_TIMER_EXPIRY.ALLOW_ON_SERVICE_FAILURE	If this flag is set to true, the UE T3501Timer Expiry request is allowed to be processed further even after all the lock acquisition retries with Bulwark Service end up with an error. If this flag is set to false, the UE T3501Timer Expiry request is responded with 500 Internal Service Error, when all the lock acquisition retries with Bulwark Service end up with an error. **Default value**: true

Table 7-36 (Cont.) Add Advanced Settings Configurations

Key	Value
CONCURRENCY.N15.TERMINATE.ALLOW_ON_SERVICE_FAILURE	If this flag is set to true, the UE Terminate request is allowed to be processed further even after all the lock acquisition retries with Bulwark Service end up with an error. If this flag is set to false, the UE Terminate request is responded with 500 Internal Service Error, when all the lock acquisition retries with Bulwark Service end up with an error. **Default value**: true
CONCURRENCY.N15.UE_NOTIFICATION.ALLOW_ON_SERVICE_FAILURE	If this flag is set to true, the UE Update Notification and UE Update Termination requests are allowed to be processed further even after all the lock acquisition retries with Bulwark Service end up with an error. If this flag is set to false, the UE Update Notification and UE Update Termination requests are responded with 500 Internal Service Error, when all the lock acquisition retries with Bulwark Service end up with an error. **Default value**: true
CONCURRENCY.N15.NOTIFICATION.ALLOW_ON_SERVICE_FAILURE	If this flag is set to true, the UE Update Notification and UE Update Termination requests are allowed to be processed further even after all the lock acquisition retries with Bulwark Service end up with an error. If this flag is set to false, the UE Update Notification and UE Update Termination requests are responded with 500 Internal Service Error, when all the lock acquisition retries with Bulwark Service end up with an error. **Default value**: true
CONCURRENCY.N15.REATTEMPT_TIMER_EXPIRY.ALLOW_ON_SERVICE_FAILURE	If this flag is set to true, the UE Reattempt Timer Expiry request is allowed to be processed further even after all the lock acquisition retries with Bulwark Service end up with an error. If this flag is set to false, the UE Reattempt Timer Expiry request is responded with 500 Internal Service Error, when all the lock acquisition retries with Bulwark Service end up with an error. **Default value**: true
CONCURRENCY.N15.CLEANUP.ENABLED	Enables concurrency feature for UE cleanup call flow. **Default value**: false
CONCURRENCY.N15.CLEANUP.LOCK_REQUEST_RETRY_COUNT	If lock acquisition failed from Bulwark due to lock request failed. **Default value**: 2
CONCURRENCY.N15.CLEANUP.LOCK_REQUEST_RETRY_BACKOFF	This is the duration for which UE service will wait once it receives ALREADY_LOCKED Response from Bulwark for the UECleanup flow. **Default value**: 1000
CONCURRENCY.N15.CLEANUP.LOCK_LEASE_DURATION	Lease duration for which lock will be acquired. After this time, Bulwark will automatically release the lock. **Default value**: 2000
CONCURRENCY.N15.CLEANUP.LOCK_WAIT_DURATION	Wait timeout for lock acquisition if not acquired. **Default value**: 3000

Table 7-36 (Cont.) Add Advanced Settings Configurations

Key	Value
CONCURRENCY.N15.CLEANUP.ALLOW_ON_SERVICE_FAILURE	If bulwark lock is not obtained after all retries for the UE Cleanup flow, then if enabled, the request will continue. If disabled, the request will be rejected. **Default value**: true

5. Click **Save**.
   The page saves the PCF UE Policy configurations.

7.3.6 PCF User Connector

This procedure provides information about configuring the PCF User Connector Service.

The **PCF User Connector** page displays the User Connector Service configurations. The page allows you to edit the configurations.

To configure the PCF User Connector service:

1. From the navigation menu under **Policy**, navigate to **Service Configurations** and select **PCF User Connector**.
   This opens the **PCF User Connector** page. The page displays the existing configurations.
2. Click Edit .
   This opens the **Edit PCF User Connector** page.
3. Check the default configuration for all the fields in all groups and edit as necessary.
   The following table describes the input fields displayed under each group:

Table 7-37 Edit PCF User Connector

Field Name	Description
System	
Server Root URL	Specifies the callback URI for notifications to be received by the User service. For example, while creating a subscription for the user with UDR.
Common	
Request Timeout	Request timeout in milliseconds for: • AmPolicyData change notification request sent to AM service • UePolicySet change notification request sent to UE service • SmPolicyData change notification request sent to SM service • SpendingLimit status change notification/ terminate request sent to SM service • On demand discovery request sent to NRF client for dynamic discovery of UDR **Default Value**: 2000 milliseconds
UDR	

Table 7-37 (Cont.) Edit PCF User Connector

Field Name	Description
Base Uri	Base Uri specifies the part of resource uri, which follows apiRoot. The Base Uri is common to all Policy Data resources. For example – In the Resource Uri `{apiRoot}/nudr-dr/{apiVersion}/policy-data/ues/{ueId}/sm-data`, `nudr-dr/{apiVersion}` is the base Uri. **Default Value**: /nudr-dr/v1
Supported Features	Refers to the value set to **supportedFeatures** field in PolicyDataSubscription while sending subscriptions request to UDR for notifying changes in policy data. **Default Value**: f
SM Data VSA Names	Indicates to provision subscriber name from Vendor Specific Attribute (VSA) data in SM Policy data for subscriber profile. **Example**: VendorSpecific-000111 **Note**: This field must be configured for Cloud Native Core PCRF deployments that have Usage Monitoring enabled that interacts with UDR to fetch Subscriber Profile Data over 3GPP N36 interface. This field indicates a list of Vendor Specific Attribute names that are provisioned in the SessionManagementPolicyData Resource on UDR that needs to be fetched by the Cloud Native PCRF Function to be used in PCRF Core and Usage Monitoring.
AM Data Uri	It refers to the section of resource uri that represents all UE related access and mobility policy attributes in the UDR for a given "ueId". AM Data Uri follows base uri part of the resource uri. **Default Value**: /policy-data/ues/{ueId}/am-data
UE Policy Set Uri	It refers to the section of resource uri that represents UE policy set attributes in the UDR for a given "ueId". UE Policy Set Uri follows base uri part of the resource uri. **Default Value**: /policy-data/ues/{ueId}/ue-policy-set
SM Data Uri	It refers to the section of resource uri that represents all PDU session related subscription attributes in the UDR for a given "ueId". SM Data Uri follows base uri part of the resource uri. **Default Value**: /policy-data/ues/{ueId}/sm-data
Usage Mon Uri	It refers to the section of resource uri that represents an individual usage monitoring resource created in the UDR and associated with a ueId and a usageMonId. Usage Mon Uri follows base uri part of the resource uri. **Default Value**: /policy-data/ues/{ueId}/sm-data/{usageMonId}

Table 7-37 (Cont.) Edit PCF User Connector

Field Name	Description
Subs To Notify Uri	It refers to the section of resource uri that represents subscriptions to notification of policy data modification. Subs To Notify Uri follows base uri part of the resource uri. **Default Value**: /policy-data/subs-to-notify
Subs To Notify Subs Id Uri	It refers to the section of resource uri that represents an individual subscription to notification of policy data modification. Subs To Notify Subs Id Uri follows base uri part of the resource uri. **Default Value**: /policy-data/subs-to-notify/{subsId}
SM Data Subscription Resource	Default value would be 1 on selection of "Sm-data" and other value is 2 on selection of "As requested by SM service".
Discover UDR with Policy as Supported Data Set	Indicates whether to send "POLICY" as data-set to be supported by the UDR during NRF discovery procedure. In turn, this configuration helps NRF filter and responds with only those UDR NF Profiles which support "POLICY" as one of the data-sets. By default, the value for this parameter is set to `false`.
Request Timeout	Specifies the timeout interval in milliseconds in which a request to UDR fails if UDR fails to respond within this value of timeout. **Default Value**: 1000 milliseconds
Enable Discovery On Demand	When this field is set to true, UDR discovery from NRF takes place for each UeId. **Default Value**: false
Retry Profile for Initial Messages	Retry Profile to be used when PCF fails to send a create message to a producer node.
Retry Profile for Subsequent Messages	Retry Profile to be used when PCF fails to send an in-session message to a producer node.
Retry Subscription Message as:	Retry Subscription Message enables the operator to choose Retry Logic for SUBSCRIBE/POST towards UDR from the follow possible options: • **Initial Message**: PCF will treat POST as Initial message and apply Initial Retry Logic for Session Retry. • **Subsequent Message**: PCF will treat POST as Subsequent message and apply Subsequent Retry Logic for Session Retry.
NF Communication Profile	The NF communication profile created for the service. This profile is created using the NF Communication Profiles page.

Table 7-37 (Cont.) Edit PCF User Connector

Field Name	Description
Send Target API Root Header in Subscribe to Notify Request	Specifies whether to include Target API Root header in the subscribe to notify request sent by PCF towards UDR. **Note**: If the user selects Model D for Policy NF Communication Model, the value for this field can be set to true or false. For other Policy NF Communication Model values, the value for this field must be set to true. By default, the value for this parameter is set to true.
Send Initial Discovery Parameters in Subscribe to Notify Request	Specifies whether to include initial discovery parameters in the subscribe to notify request sent by PCF towards UDR. By default, the value for this parameter is set to false.
Send Producer Id in Discovery Header in Subscribe to Notify Request	Specifies whether to include Producer ID in discovery header in the subscribe to notify request sent by PCF towards UDR. By default, the value for this parameter is set to false.
CHF	
Retry Profile for Initial Messages	Retry Profile to be used when PCF fails to send a create message to a producer node.
Retry Profile for Subsequent Messages	Retry Profile to be used when PCF fails to send an in-session message to a producer node.
NF Communication Profile	The NF communication profile created for the service. This profile is created using the NF Communication Profiles page.
PCF Service name in Binding Header	Specifies the custom/3GPP defined PCF service name to be included in the binding header in the subscription request or notification responses sent towards CHF. Possible values: • Aggregate or custom service name that represents a group of PCF services with aggregated service level information. For example: `npcf-custom-service` • Any of the 3GPP defined PCF service names such as `npcf-smpolicycontrol` or `npcf-am-policy-control` There is no default value configured for this field.

> **Note**
>
> - If both **Retry Profile for Initial Messages** and **Retry Profile for Subsequent Messages** are not configured (default case) no retry is attempted.
> - If both **Retry Profile for Initial Messages** and **Retry Profile for Subsequent Messages** are configured, retries is attempted accordingly.
> - If **Retry Profile for Initial Messages** is configured but **Retry Profile for Subsequent Messages** is not configured then the profile for initial messages is used for subsequent messages. (to be backward compatible with release 1.8.x)
> - If **Retry Profile for Initial Messages** is not configured but **Retry Profile for Subsequent Messages** is configured, then retry is not attempted for initial messages. But it is attempted for the subsequent messages.
> - If **Send PCF Service Name in Binding Header** parameter under **NF Bindings Setting** section in **NF communication profile** page is enabled for CHF interface , the aggregate custom Service name or a 3gpp defined pcf service name must be defined in **PCF Service name in Binding header** parameter. This service name will be included in the binding header towards CHF sent over N28/Nchf interface.

4. Perform the following steps to configure **Advanced Settings** that sets the key precendence on user connector:

   a. Click the Add
   The page opens the **Add Advanced Settings** dialog box.

   b. In the dialog box, enter the following **key** and respective **value**:

Table 7-38 Parameters for Advanced Settings

Keys	Value
UDR.KeyPrecedence	It is used to set the Key Precedence given on User Connector. **Default Value**: SUPI, GPSI
CONCURRENCY_LOCK_FAILURE_ERROR_CODE	Indicates an error in PDS Service in acquiring a lock from Bulwark Service to process a notification from UDR or CHF. **Default Value**: 500
RETAIN_ORIGINAL_MONITORED_RESOURCE_URI_ON_REVALIDATION	If the flag is set to true the monitoredResourceUri sent in the POST will be retained for the subsequent PUT request. **Default Value**: false
UDR_errorHandlerEnabled	This flag enables error handling configuration with UDR connector service. **Default Value**: false
CHF_errorHandlerEnabled	This flag enables error handling configuration with CHF connector service. **Default Value**: false

Table 7-38 (Cont.) Parameters for Advanced Settings

Keys	Value
UDR_NF_PROFILE_COOKIE_ENABLED	This flag indicates whether the UDR profiles discovered during UDR-GET needs to be cached till UDR-POST is completed (success/ failure). **Default Value**: false
UDR_NF_PROFILE_COOKIE_COMPRESS	This flag indicates whether to compress the UDR profiles cached between GET and POST request. **Default Value**: false
UDR_NF_PROFILE_COOKIE_LIMIT	This flag indicates the number of UDR profiles between GET and POST that can be cached. **Default Value**: 10 **Note:** It is recommended to enable UDR_NF_PROFILE_COOKIE_COMPRESS to true so that we can support caching of upto 10 UDR profiles. For supporting lesser number of UDR profile caching (like 3-4), we can disable the compression. Caching of more than 10 profiles is not supported.
UDR_GET_USE_RELATED_RESOURCE	This flag allows GET for related resources (for example, amPolicyData and uePolicySet) from the same UDR where previous GET was done. **Default Value**: false

5. Click **Save**.
   The page saves the PCF User Connector configurations.

7.3.7 Configuring Usage Monitoring

This procedure provides information about managing configurations for Usage Monitoring service.

The Usage Monitoring page allows you to edit default Usage Monitoring configurations.

To configure, perform the following steps:

1. From the navigation menu, under **Policy**, click **Service Configurations**, and select **Usage Monitoring**.
   This opens the Usage Monitoring page with the default configurations.
2. Click **Edit**.

   This opens the Edit Usage Monitoring page.
3. Enter values for the input fields, described in the following table:

Table 7-39 Usage Monitoring Configurations

Field Name	Description
Enable PRE	Specifies whether to enable or disable interaction with PRE for Policy evaluation as per the match list.

Table 7-39 (Cont.) Usage Monitoring Configurations

Field Name	Description
Minimum Volume Grant (bytes)	Enter the minimum grant value (in bytes) that can be approved for volume based usage monitoring. If the grant value published by PRE is less than the value for this field, then Policy considers the minimum grant value out of the two. After deducting the current grant, if the remaining grant value is less than the minimum grant value, then the remaining grant is also added to the current grant. **Default value**: 2048 **Range**: 512 - 1048576 (1MB)
Maximum Volume Grant (bytes)	Enter the maximum grant value (in bytes) that can be approved for volume based usage monitoring. If the grant value published by PRE is greater than the value for this field, then Policy considers the maximum grant value out of the two. **Default value**: 2048 Range: 1024 - 107374182400 (100 GB)
Minimum Time Grant (seconds)	Enter the minimum grant value (in seconds) that can be approved for time based usage monitoring. If the grant value published by PRE is less than the value for this field, then Policy considers the minimum grant value out of the two. After deducting the current grant, if the remaining grant value is less than the minimum grant value, then the remaining grant is also added to the current grant. **Default value**: 300 Range: 30 - 3600 (1 hour)
Maximum Time Grant (seconds)	Enter the maximum grant value (in seconds) that can be approved for time based usage monitoring. If the grant value published by PRE is greater than the value for this field, then Policy considers the maximum grant value out of the two. **Default value**: 300 Range: 60 - 86400 (1 hour)
Inactivity Time (seconds)	Enter the time interval (in seconds) during which if no packets are received for a given monitoring key, PGW or SMF stops measuring the time. **Default value**: 86400 Range: 0 - 604800 (7 Days)

Table 7-39 (Cont.) Usage Monitoring Configurations

Field Name	Description
Usage Accumulation Start Date and Time	Select the start time and date from when the Usage Monitoring service starts counting the usage reset periodicity. The values are: • **Plan Start Date and Time**: Counted from the plan start date and time as provisioned on the UDR. • **Plan Activation Date and Time**: Counted from the plan activation date and time as provisioned on the UDR. • **Period Start Date and Time**: Counted from the period start date and time.
Week Start Day	Enter the starting day of the week. **Default value**: Sunday
Default Data Rollover Profile	Indicates the default data rollover profile to use for UDR provided data limits for which rollover is applicable. That is, "Data Rollover" attribute is set to "true" and a separate "Data Rollover Profile" attribute is not provided.
Enable Quota Migration	Indicates whether to enable retrieval of OCPM data from CnUDR. If enabled, CnPCRF interacts with CnUDR to collect the quota details of the OCPM (4G) subscribers. **Default value**: false **Note**: Disabling this flag while Subscriber Data Migration from OCUDR (4G) to CnUDR (5G) is incomplete may result in Quota loss. It is recommended to not fallback to OCPM PCRF once this flag is disabled as after this, the data provided to OCPM PCRF may NOT be up to date.
Reverse Priority	Indicates the priority consideration order. When this flag is enabled: Higher number indicates lower priority When this flag is disabled: Higher number indicates higher priority **Default value**: `true`
Enable Pro-rated Data at The Time of Activation	Iindicates whether pro-rated data calculation should be applied to the selected plan at a time of activation or not. **Default value**: false
Enable Pro-rated Data at The Time of Billing Day Change	Iindicates whether pro-rated data calculation should be applied to the selected plan at a time obilling day change . **Default value**: false
Apply Billing Day / Data Plan Change	Indicates if the flexible billing cycle option is enabled. The accepted values are: • Current Billing Cycle • Next Billing Cycle **Default value**: Next Billing Cycle

Table 7-39 (Cont.) Usage Monitoring Configurations

Field Name	Description
Data Compression Scheme	Used to configure the data compression in UmContext table. This flag accepts the following values: • `Disabled`: Indicates that the data will not be compressed and will be stored in the column in uncompressed format. The value of `COMPRESSION_SCHEME` for that row is set to the default value 0. • `MySQL Compressed`: Indicates that the data will be compressed using MySQL and will be stored in the column in compressed format. The value of `COMPRESSION_SCHEME` for that row is set to 1. • `Application Compressed`: Indicates that the data will be compressed using Zlib and will be stored in the column in compressed format. The value of `COMPRESSION_SCHEME` for that row is set to 2. **Default value**: Disabled When accessing the data from `UmContext .v`, the application determines the compression status based on the value of `COMPRESSION_SCHEME` and does the decompression accordingly. For more details on data compression, see *Data Compression in Usage Monitoring* section in Usage Monitoring on Gx Interface.
Default Volume Grant	
Grant Unit	Select a value from the drop-down list: • **Percent**: Grant volume is calculated as a percentage. • **Bytes**: Grant volume is calculated as an absolute value. **Default value**: Percent
Grant Value (Percent)	**Note**: This field becomes active when you select **Grant Unit** as percent. Enter the default Grant Volume in percentage, which is considered when PRE is not invoked or PRE does not return a decision. This value is applied to the Total Volume and divided equally among the Uplink and Downlink Volumes when available. **Default value**: 10 **Range**: 1-100 When the grant value is 0 and Grant is not calculated by PRE, then volume based Usage Monitoring is disabled for the session.

Table 7-39 (Cont.) Usage Monitoring Configurations

Field Name	Description
Grant Value (Bytes)	**Note**: This field becomes active when you select **Grant Unit** as Bytes. Enter the default Grant Volume in Bytes, which is considered when PRE is not invoked or PRE does not return a decision. This value is applied to the Total Volume and divided equally among the Uplink and Downlink Volumes when available. **Default value**: 2048 **Range**: 512 - 1048576 (1 MB) When the grant value is 0 and Grant is not calculated by PRE, then volume based Usage Monitoring is disabled for the session.
Grant Source	**Note**: This field becomes active when you select **Grant Unit** as Percent. Select a value from the drop-down list: • **Initial Volume**: Calculate grant volume from the initial volume. • **Used Volume**: Calculate grant volume from the used volume. • **Remaining Volume**: Calculate grant volume from the remaining volume. **Default value**: Initial Volume
Default Time Grant	
Grant Unit	Select a value from the drop-down list: • **Percent**: Grant time is calculated as a percentage. • **Seconds**: Grant time is calculated as an absolute value. **Default value**: Percent
Grant Value (Percent)	**Note**: This field becomes active when you select **Grant Unit** as percent. Enter the default Grant time in percentage, which is considered when PRE is not invoked or PRE does not return a decision. **Default value**: 10 **Range**: 1-100 When the grant value is 0 and grant is not calculated by PRE, then time based Usage Monitoring is disabled for the session.
Grant Value (seconds)	**Note**: This field becomes active when you select **Grant Unit** as seconds. Enter the default Grant time in seconds, which is considered when PRE is not invoked or PRE does not return a decision. **Default value**: 3600 **Range**: 30 - 86400 (1 day) When the grant value is 0 and grant is not calculated by PRE, then time based Usage Monitoring is disabled for the session.

Table 7-39 (Cont.) Usage Monitoring Configurations

Field Name	Description
Grant Source	**Note**: This field becomes active when you select **Grant Unit** as Percent. Select a value from the drop-down list: • **Initial Duration**: Calculate grant time from the initial duration. • **Used Duration**: Calculate grant time from the used duration. • **Remaining Duration**: Calculate grant time from the remaining duration. **Default value**: Initial Volume
Default Usage Levels	
Minor Usage Level (%)	Enter threshold value in percentage indicating minor usage. **Default value**: 50
Major Usage Level (%)	Enter threshold value in percentage indicating major usage. **Default value**: 75
Critical Usage Level (%)	Enter threshold value in percentage indicating critical usage. **Default value**: 95
Exhausted Usage Level (%)	Enter threshold value in percentage indicating usage level has exhausted allowed limit. **Default value**: 100
Custom Attribute Mapping	
Reset Day & Time	Enter a string identifying the custom attribute in UDR response that represents the usage reset day. **Example**: `SmPolicyData/umDataLimits/{limitId}/billingDay`
Data Plan Name	To map the custom attribute available in the UDR provided data limit to name. A "name" associated with the data limit can be used in a Selection Profile to select the plan to apply when multiple active plans are found. **Default vlaue**: SmPolicyData/umDataLimits/{limitId}/name
Data Plan Priority	To map the custom attribute available in the UDR provided data limit to priority. A "priority" associated with the data limit would help to select the plan to apply when multiple active plans are found. Possible values allowed in this field in UDR Data Limit are: any integer. **Default value**: SmPolicyData/umDataLimits/{limitId}/priority

Table 7-39 (Cont.) Usage Monitoring Configurations

Field Name	Description
Data Plan Type	To map the custom attribute available in the UDR provided data limit to type. A "type" associated with the data limit would help to order the selection of the plan. Possible values are: "base", "pass", "top-up". A separate configuration would order the selection, for example: Pass > Base Plan > Top-up. Default: SmPolicyData/umDataLimits/{limitId}/type Default value: base **Note**: The Plan Type, Priority within that Plan Type and the Selection Order of Plan Types together decide the order of selection of Data Plans.
Data Rollover	To map the custom attribute available in the UDR provided data limit to indicate whether rollover is enabled for this data limit. Possible values allowed in this field in UDR Data Limit are: "true" and "false". **Default value**: SmPolicyData/umDataLimits/{limitId}/rollover Default value to apply if this mapped attribute is absent: false
Data Rollover Profile	To map the custom attribute available in the UDR provided data limit to a pre-configured rollover profile. See "Data Rollover Profile" for more information. Possible values allowed in this field in UDR Data Limit are: A "Data Rollover Profile" name pre-configured on Policy. **Default value**: SmPolicyData/umDataLimits/{limitId}/rolloverProfile
Parent Plan Name	To map the custom attribute available in the UDR provided data limit to parent plan name. Attributes not present in the UmDataLimit shall be picked up from the "parent" (if present). The parent plan name may be a configured Data Limit Profile Name. **Default value**: SmPolicyData/umDataLimits/{limitId}/parent
Parent Plan Source	To map the custom attribute available in the UDR provided data limit to parent plan source. A parent plan source "parent" (if present). The parent plan name may be a configured Data Limit Profile Name. Possible values allowed in this field in UDR Data Limit are: "data-limit-profile", "um-data-limit". **Default value**: SmPolicyData/umDataLimits/{limitId}/parentSource Default value to apply if this mapped attribute is absent: "data-limit-profile"

4. To add selection profles for the **Data Limit Selection**, perform the following configurations:

Field Name	Description
Default Data Limit Profile	Indicates the default data limit profile to apply when no (usable) data limits were obtained from UDR and no (usable) data limits were applied by policy.
Default Data Limit Selection Profile	Indicates the default data limit selection profile to use while selecting a data limit among more than one active data limits provided by UDR.
Default Data Limit Sorting Profile	Indicates the default data limit sorting profile to use while selecting a data limit among more than one active data limits provided by UDR.
Selection Order 1	Selection Order determines the order/priority of selection of different types of plans. Default Value of Selection Order 1 : Pass
Selection Order 2	Default Value of Selection Order 2 : Base Plan
Selection Order 3	Default Value of Selection Order 3: Top-Up

5. Enter values for the input fields, described in the following table:

Table 7-40 Conflict Resolution Configurations

Field Name	Description
Enable ETag / If-Match headers	Indicates if the ETag functionality is enabled or disabled between PDS and UDR. By default, this is disabled. Following scenarios may occur if it is enabled: • When the Usage Monitoring service sends a GET request for the first time, then PDS will fetch latest ETag details from UDR and update it in THE pdssubscriber table. • when the Usage Monitoring service sends PATCH request to PDS, PDS sends the request to UDR using if-match header. UDR matches the ETag of PDS with UDR. If it matches, then UDR replies with the latest ETag that gets udpated on PDS. • If the If-Match headers failS, it means ETag on PDS does not match with UDR. UDR replies with 412 error response and based on the response code, Usage Monitoring service sends a force request to PDS to get the latest ETag from UDR and update it on PDS database.
Maximum Number of Conflict Resolution Attempts	Specifies the number of retry the Usage Monitoring service will do when if-match will fail on UDR. Default value: 5

6. Enter values for as described in the following table to configure audit cycles for Usage Monitoring Gx sessions:

Table 7-41 Audit Configurations

Field Name	Description
Enable	Used to enable or disable the Audit registration for Usage Monitoring service. When this field is enabled, Usage Monitoring sends registration request to Audit service. When this field is disabled, Usage Monitoring sends de-registration request to Audit service. **Default value**: true
Notification Rate (per second)	Indicates the number of stale records that Audit service notifies to Usage Monitoring service in one second. Value of this parameter ranges between 20 and 700. **Default and Recommended Value**: 50 **Note**: To configure higher number than the recommended value, contact My Oracle Support (https://support.oracle.com)
Policy Association Age (in minutes):	Indicates the age of a Usage Monitoring session, after which the session is considered to be stale and the PCRF Core service is queried for the presence of such associations. **Note**: This value should be higher than the Gx session audit. Value of this parameter ranges between 1 and 10080. **Default Value**: 1440
Policy Association Maximum Age (in minutes):	Indicates the maximum age of a UMPolicyAssociation, after which the session is purged from UMContext database without sending further queries. **Note**: This value should be higher than the Gx session audit. Value of this parameter ranges between 1 and 20160. **Default Value**: 2880
Minimum Audit Attempts	Indicates the minimum number of queries to PCRF Core service before deleting a session. **Range**: 0-255 **Default Value**: 0
Minimum Audit Passes Interval (in minutes)	Defines the time when next audit for the Usage Monitoring service table is done after delta time if auditing this table has been finished before this specified time. **Default Value**: 330

7. Enter values for the input fields, described in the following table to configure message priorities for Congestion Control in Usage Monitoring:

Table 7-42 Message Default Priority for Congestion Control Configurations

Field Name	Description
UM Session Create API	Set request priority for UM session creation (CCR-I) request. **Allowed Values**: 0-31 **Default Value**: 20
UM Session Update API	Set request priority for UM session update (CCR-U) request. **Allowed Values**: 0-31 **Default Value**: 17
UM Session Terminate API	Set request priority for UM session terminate (CCR-T) request. **Allowed Values**: 0-31 **Default Value**: 15
UM Session Notify API	Set request priority for UM session notification request. **Allowed Values**: 0-31 **Default Value**: 17
UM Session Audit Subscriber API	Set request priority for UM session audit subscriber request. **Allowed Values**: 0-31 **Default Value**: 31
UM Session Search Subscriber API	Set request priority for UM session search for subscriber request. **Allowed Values**: 0-31 **Default Value**: 30
UM Session Audit Notify API	Set request priority for terminate session (CCR-T) request. **Allowed Values**: 0-31 **Default Value**: 31
Congestion Error Code	Configure the response code of the any rejected requests by Usage Monitoring due to congestion state. By default, Usage Monitoring sends 503 as response code. If configured then it shall be 5xx only. **Allowed Values**: 5xx **Default Value**: 503

8. Perform the following steps to configure **Advanced Settings** for Usage Monitoring:

   a. Click the ⊕ Add
   The page opens the **Add Advanced Settings** dialog box.

   b. In the dialog box, enter the following **keys** and respective **values**:

The following table describes the keys and values:

Table 7-43 Add Advanced Settings Configurations

Key	Value
PREVIOUS_UM_CONTEXT_TABLE_SLICING_COUNT	Used to configure the previous slicing count for UMContext table. If `UM_CONTEXT_TABLE_SLICING_COUNT` Helm parameter is changed and the Usage Monitoring service cannot find the data with the current slicing count, it uses `PREVIOUS_UM_CONTEXT_TABLE_SLICING_COUNT` to search for the data. If there are multiple values for `PREVIOUS_UM_CONTEXT_TABLE_SLICING_COUNT`, the Usage Monitoring service uses the latest previous count and if the data is still not found, it uses the next previous count. That is, if `PREVIOUS_UM_CONTEXT_TABLE_SLICING_COUNT` contains two values such as 0,5, the Usage Monitoring service first uses the previous count as 5 to search for the data. If it is still not found, the service uses the previous count as 0 to search for the data. **Default value**: 0

9. Click **Save** to save Usage Monitoring service configurations.

7.3.8 PCRF Core Service Configurations

This section describes how to configure the PCRF Core service.

To access PCRF Core service configurations from CNC Console home page, expand **Policy**, navigate to **Service Configurations** and select **PCRF Core**.

On clicking PCRF Core, you can select to customize any of the following configurations:

- Settings
- Serving Gateway
- Network Element

7.3.8.1 Settings

This procedure provides information about configuring the PCRF Core settings.

The **Settings** page under **PCRF Core** displays the PCRF Core service settings. The page allows you to edit the configurations.

To edit the PCRF Core settings:

1. From the navigation menu under **Policy**, navigate to **Service Configurations**, click **PCRF Core**, and select **Settings**.
   This opens the **Settings** page. The page displays the existing settings for PCRF Core service.
2. Click Edit .
   This opens the **Edit Settings** page.
3. Expand the **User** group, to enable or disable switches, described in the following table:

Table 7-44 User Group Settings

Field Name	Description
Validate User	If the User profile, for any of the User Data types given under the **User Data Types** group, are fetched successfully, the user is considered to be a known user. However, if the user profile lookup has failed on all the configured data sources, the user profile is marked as Unknown. If this switch is enabled and user is marked as Unknown, the session creation requests are rejected. If this switch is disabled and user is marked as Unknown, the session creation requests are handled and sessions are created successfully.
Enable Subscriber Variables	If this switch is enabled, PCRF Core controls the querying and saving of remote subscriber state variables (SSV) on PDS. When CCR message is received by PCRF Core, then it queries PDS to fetch SSV for the user, instead of retrieving it locally. By default, the switch remains disabled.
User Data Types	
SmPolicyData	If this switch is enabled, PCRF-Core fetches SMPolicyData from nUDR. To configure the SMPolicyData attributes, see SmPolicyData Attributes.
Operator Specific Data	If this switch is enabled, PCRF fetches OperatorSpecificData (imported using Custom Schema) from nUDR. To configure the SMPolicyData attributes, see Table 7-46.
Ldap Data	If this switch is enabled, PCRF-Core fetches user profile from LDAP through PDS.
OCS Spending Limit	If this switch is enabled, PCRF-Core fetches policy counters for a subscriber from 4G OCS. Irrespective of whether OCS is configured as primary datasource or on-demand datasource, this switch must be enabled to retrieve policy counters. To configure the OCS spending limit attributes, see Table 7-47.
Sh Data	If this switch is enabled, PCRF-Core fetches user profile from 4G UDR. **Note**: Sh interface is not supported for Converged Policy mode of deployment.

The following table describes the **SmPolicyData** switches available in the **Attributes** column:

Table 7-45 SmPolicyData Attributes

Attribute Name	Description
Subscribe to Notify	When this switch is enabled, PCRF Core subscribes with the UDR to get notified on changes in subscriber profile. **Note**: When this switch is disabled, Usage Monitor feature cannot be tested. It will not be functional. By default, this option is enabled.
Query User on Update	Determines if user query from UDR on update is enabled. When this option is enabled, PCRF Core queries the UDR about the subscriber present in the CCR request or GX message by sending a GET request for “SmPolicyData” resource on the nudr-dr service. **Note**: If `Subscribe To Notify` is enabled for PCRF Core, then PCRF services will not query the UDR during a Update requests. The PDS caches the subscriber profile when the “Subscribe To Notify” option is enabled, in that case, the PCRF core may not always reach the UDR when the subscriber profile is found in the local cache. By default, this option is disabled.
Query User on Terminate	Determines if user query from UDR on delete is enabled. When this option is enabled, PCRF Core queries the UDR about the subscriber present in the CCR request or the GX message by sending a GET request for “SmPolicyData” resource on the nudr-dr service. **Note**: If `Subscribe To Notify` is enabled for PCRF Core, then PCRF service will not query the UDR during a Terminate requests. The PDS caches the subscriber profile when the “Subscribe To Notify” option is enabled, in that case, the PCRF Core may not always reach the UDR when the subscriber profile is found in the local cache. By default, this option is disabled.
Query User on Reauth	Determines if user query from UDR on reauth is enabled. When this option is enabled, PCRF Core queries the UDR about the subscriber, when it receives a Reauthorization request, such as Rx or Policy Authorization request by sending a GET request for “SmPolicyData” resource on the nudr-dr service. **Note:** The PDS caches the subscriber profile when the “Subscribe To Notify” option is enabled, in that case, the PCRF Core may not always reach the UDR when the subscriber profile is found in the local cache. By default, this option is disabled.
Auto Enrollment on UDR	To specify whether to enable or disable the autoprovisioning of the subscriber profile at UDR when the subscriber profile is not available.

Table 7-45 (Cont.) SmPolicyData Attributes

Attribute Name	Description
Default-S-NSSAI	To configure a default S-NSSAI, which will be used when a subscriber is auto-enrolled on the UDR. This parameter is applicable only when Auto-Enrollment on UDR is set to `true`. The value configured here must match the Default S-NSSAI configured on UDR.

The following table describes the **Operator Specific Data** switch available in the **Attributes** column:

Table 7-46 Operator Specific Data Attributes

Attribute Name	Description
Subscribe to Notify	When this switch is enabled, PCRF Core subscribes to get notified of any changes in the profile. By default, this option is true.

The **OCS Spending Limit** switches available in the **Attributes** column:

Table 7-47 OCS Spending Limit Attributes

Attribute Name	Description
Async Query	When this switch is enabled, PCRF Core performs on-demand lookup of Policy Counters for a subscriber using the **Fetch Policy Counters** from *OCS* policy action. By default, this option is disabled.
Enable force lookup on Update	When this switch is enabled, the core services (PCRF Core and SM) sends OCS request skipping the PDS optimiziations. By default, this option is disabled.

4. Under the **Usage Monitoring** group, update the values for the fields, described in the following table:

Table 7-48 Usage Monitoring Settings

Field Name	Description
Enabled	Specifies whether to enable or disable Usage Monitoring service for PCRF. By default, this switch is `disabled`.
APN List	Specifies the list of APNs for which usage monitoring is enabled. If you leave this field empty, Usage Monitoring is enabled for all APNs.
Attribute Forwarding	

Table 7-48 (Cont.) Usage Monitoring Settings

Field Name	Description
Interface Type	Indicates the interface type for which the profile applies. For more details on Attribute Forwarding Profiles, see *Attribute Forwarding Profiles* section in Usage Monitoring on Gx Interface.
Message Type	Indicates the message type for which the profile applies.
Forwarding Profile	Refers to the configured Attribute Forwarding Profile.

5. Under **MCPTT** group, apply the configurations to set the default ARP and default preemption control.

Table 7-49 ARP Settings

Parameter	Description
Default ARP Settings	
Priority Level	Indicates the priority level. Range: 1-15 Default Value: 1
Preemption Capability	Indicates if the preemption cabaility is enabled or disabled. It can take the following values: • `PREEMPTION_CAPABILITY_ENABLED` • `PREEMPTION_CAPABILITY_DISABLED` Default Value: `PREEMPTION_CAPABILITY_ENABLED`
Preemption Vulnerability	Indicates if the preemption vulnerability is enabled or disabled. It can take the following values: • `PREEMPTION_VULNERABILITY_ENABLED` • `PREEMPTION_VULNERABILITY_DISABLED` Default Value: `PREEMPTION_VULNERABILITY_DISABLED`
Default Preemption Control	
Default Preemption Control info	Includes the Default Preemption Control information: • `LEAST_RECENT_ADDED_FLOW` • `MOST_RECENT_ADDED_FLOW` • `HIGHEST_BANDWIDTH_FLOW` Default Value: `LEAST_RECENT_ADDED_FLOW`

6. Expand the **Audit** group, and update the values for the fields described in the following table:

Table 7-50 Audit Group Settings

Field Name	Description
Enabled	Determines whether to send registration request to Audit service or not. By default, this switch is `enabled`.
Notification Rate (per second)	Defines the number of stale records which Audit service notifies to PCRF Core service in one second. **Default and Recommended Value**: 50 **Note**: To configure higher number than the recommended value, contact My Oracle Support (https://support.oracle.com)
Policy Association Age (in minutes) for Gx Session	Defines the age of a Gx session after which a record is considered to be stale on PCRF Core and the PGW is queried for presence of such associations. **Default Value**: `1440`
Policy Association Maximum Age (in minutes) for Gx Session	Defines the maximum age of a Gx session after which a record is purged from PCRF Core database (Gx Session table) without sending further queries to PGW. **Default Value**: `2880`
Minimum Audit Attempts for Gx session	Specifies the minimum number of consecutive failed audit attempts until maxTTL / forceTTL is reached. If maxTTL is reached and audit_attempts >= Minimum Audit Attempts for maxTTL, Audit service sends notification to PCRF Core service with maxTTL flag set to *true*. PCRF Core service triggers the TERMINATE (CCR-T) leg. **Range**: 0-255 **Default Value**: 0 **Note**: If maxTTL is not reached and if audit attempts are reached, the number of audit attempts are incremented until maxTTL is reached.
Policy Association Age (in minutes) for Rx Session	Defines the age of an Rx session after which a record is considered to be stale on PCRF Core. When the **maxTTL** flag is **TRUE**, Rx sessions are cleaned. **Note**:The cleanup functionality is not implemented for Rx sessions, if the**maxTTL** flag is **false** **Default Value**: `1440`
Policy Association Maximum Age (in minutes) for Rx Session	Defines the maximum age of an Rx session after which a record is purged from PCRF Core database (Rx Session table) without sending further queries to PGW. **Default Value**: `2880`
Policy Association Age (in minutes) for Sd Session	Defines the age of an Sd session after which a record is considered to be stale on PCRF Core. **Default Value**: `1440`

Table 7-50 (Cont.) Audit Group Settings

Field Name	Description
Policy Association Maximum Age (in minutes) for Sd Session	Defines the maximum age of an Sd session after which a record is purged from PCRF Core database (Sd Session table) without sending further queries to PGW. **Default Value**: `2880`
Minimum Audit Attempts for Sd session	Specifies the minimum number of consecutive failed audit attempts until maxTTL / forceTTL is reached. If maxTTL is reached and audit_attempts >= Minimum Audit Attempts for maxTTL, Audit service sends notification to PCRF Core service with maxTTL flag set to *true*. PCRF Core service triggers the TERMINATE (CCR-T) leg. **Range**: 0-255 **Default Value**: 0 **Note**: If maxTTL is not reached and if audit attempts are reached, the number of audit attempts are incremented until maxTTL is reached.
Minimum Audit Passes Interval (in minutes)	Defines the time when next audit for the PCRF Core service table begins after delta time if auditing this table has been finished before this specified time. **Default Value**: `330`

7. Enter the values for the input fields, available under the **Pending Transaction** group. The following table describes the fields:

Table 7-51 Edit Pending Transaction Configurations

Field Name	Description
Enabled	Specifies whether to enable or disable Pending Transaction. By default, this switch is `disabled`.
RAR Retry Backoff (in milliseconds)	Indicates the time gap between the consecutive retry RAR messages when PGW is sending error response with cause of DIAMETER_PENDING_TRANSACTION and RAR message is triggered by PDS/AF/PRE. **Default Value**: `1000`
RAR Retry Count	Indicates the count for RAR messages when PGW is sending error response with cause of DIAMETER_PENDING_TRANSACTION and RAR message is triggered by Notification message from PDS/AF/PRE. **Default Value**: 2
User Notify Reauth Error Handling	Indicates if an RAR initiated by User Notification message need to be discarded or retried when it is responded by P-GW with an error of DIAMETER_PENDING_TRANSACTION. **Default Value**: RETRY

Table 7-51 (Cont.) Edit Pending Transaction Configurations

Field Name	Description
AF Reauth Error Handling	Indicates if an RAR initiated by AF message should be discarded or retried when it is responded by PGW with an error of DIAMETER_PENDING_TRANSACTION. **Default Value**: RETRY
Reauth Error Handling:	Indicates if an RAR initiated by any message should be discarded or retried when it is responded by PGW with an error of DIAMETER_PENDING_TRANSACTION. **Default Value**: RETRY

8. Perform the following steps to configure Advanced Settings that allows PCRF Core to handle concurrent requests:

   a. Click the ⊕ Add
   The page opens the **Add Advanced Settings** dialog box.

   b. In the dialog box, enter the following **keys** and respective **values**:

The following tables describe the keys and values:

Table 7-52 General

Key	Default Value
CONCURRENCY.ENABLED	false
Gx.DisableSupportedFeatures	false
Rx.DisableSupportedFeatures	false
DISTRIBUTE_GX_TRAFFIC_USING_TABLE_SLICING	When this key is enabled, traffic for GxSessions are distributed across all the new tables. PCRF Core service will register the new tables with Audit Service and also distributes the new sessions to the new slices. **Default value**: false
USER.ocsSpendingLimit.resetContextOnGxCreate	When the value of this key is set to `true`, PCRF Core sets `resetContext` to `true` in PDS request reqParam for the first Gx session and PDS will revalidate the Sy session. **Default value**: false
USER.SEQUENTIAL_SSV_UPDATE_AND_TERMINATE	When enabled, if the request triggers SSV_UPDATE towards PDS and the response has not yet been received, then the attempt to send delete request towards PDS will wait until SSV_UPDATE is done or timed out. When disabled, PDS delete will not check if SSV_UPDATE is still in progress by the same request. **Default value**: false

Table 7-53 Add Advanced Settings Configurations for Gx CCR-I

Key	Default Value
CONCURRENCY.GX.CREATE.LOCK_REQUEST_RETRY_COUNT	2
CONCURRENCY.GX.CREATE.LOCK_REQUEST_RETRY_BACKOFF	1000
CONCURRENCY.GX.CREATE.LOCK_LEASE_DURATION	2000
CONCURRENCY.GX.CREATE.LOCK_WAIT_DURATION	3000
CONCURRENCY.GX.CREATE.ENABLED	True
CONCURRENCY.GX.CREATE.ALLOW_ON_SERVICE_FAILURE	True
USER.ocsSpendingLimit.resetContextOnGxCreate	false When the value of this key is true, the resetContext flag is set to true in PDS request reqParam for the first Gx session and PDS will revalidate the sy session.
USER.ocsSpendingLimit.createContextOnFailure	If this flag is enabled the PCRF Core sends this as a request parameter toward PDS. PDS creates a dummy context for OCS in case of getting error while performing OCS lookup. This will support the PDS when it receives notificaiton from UDR to fetch OCS data. This is applicable for CCR-I and CCR-U messages with Asynchronous flow either set as true or false. **Default Value**: false
USER.smPolicyData.createContextOnFailure	If this flag is enabled the PCRF Core sends this as a request parameter toward PDS. At PDS, if it receives an error during SmPolicyData-UDR lookup, then PDS creates a dummy context for SmPolicyData. This creation of dummy context helps, when PDS receives next request for a UDR lookup and this time it is successful, then both the sessions will have the latest SmPolicyData. This is applicable for CCR-I and CCR-U Messages too. **Default value**: false
USER.operatorSpecificData.createContextOnFailure	If this flag is enabled the PCRF Core sends this as a request parameter toward PDS. At PDS, if it receives an error during operatorSpecificData-UDR lookup, then PDS creates a dummy context for operatorSpecificData. This creation of dummy context helps, when PDS receives next request for a UDR lookup and this time it is successful, then both the sessions will have the latest operatorSpecificData. This is applicable for CCR-I and CCR-U Messages too. **Default value**: false

Table 7-54 Add Advanced Settings Configurations for Gx CCR-U

Key	Default Value
CONCURRENCY.GX.MODIFY.LOCK_REQUEST_RETRY_COUNT	2
CONCURRENCY.GX.MODIFY.LOCK_REQUEST_RETRY_BACKOFF	1000
CONCURRENCY.GX.MODIFY.LOCK_LEASE_DURATION	2000
CONCURRENCY.GX.MODIFY.LOCK_WAIT_DURATION	3000
CONCURRENCY.GX.MODIFY.ENABLED	True
CONCURRENCY.GX.MODIFY.ALLOW_ON_SERVICE_FAILURE	True

Table 7-55 Add Advanced Settings Configurations for Gx CCR-T

Key	Default Value
CONCURRENCY.GX.DELETE.LOCK_REQUEST_RETRY_COUNT	2
CONCURRENCY.GX.DELETE.LOCK_REQUEST_RETRY_BACKOFF	1000
CONCURRENCY.GX.DELETE.LOCK_LEASE_DURATION	2000
CONCURRENCY.GX.DELETE.LOCK_WAIT_DURATION	3000
CONCURRENCY.GX.DELETE.ENABLED	True
CONCURRENCY.GX.DELETE.ALLOW_ON_SERVICE_FAILURE	True

Table 7-56 Add Advanced Settings Configurations for Gx RAR

Key	Default Value
CONCURRENCY.GX.REAUTH.LOCK_REQUEST_RETRY_COUNT	2
CONCURRENCY.GX.REAUTH.LOCK_REQUEST_RETRY_BACKOFF	1000
CONCURRENCY.GX.REAUTH.LOCK_LEASE_DURATION	2000
CONCURRENCY.GX.REAUTH.LOCK_WAIT_DURATION	3000
CONCURRENCY.GX.REAUTH.ENABLED	True
CONCURRENCY.GX.REAUTH.ALLOW_ON_SERVICE_FAILURE	True

Table 7-57 Add Advanced Settings Configurations for SD TSR

Key	Default Value
CONCURRENCY.SD.CREATE.LOCK_REQUEST_RETRY_COUNT	2

Table 7-57 (Cont.) Add Advanced Settings Configurations for SD TSR

Key	Default Value
CONCURRENCY.SD.CREATE.LOCK_REQUEST_RETRY_BACKOFF	1000
CONCURRENCY.SD.CREATE.LOCK_LEASE_DURATION	2000
CONCURRENCY.SD.CREATE.LOCK_WAIT_DURATION	3000
CONCURRENCY.SD.CREATE.ENABLED	True
CONCURRENCY.SD.CREATE.ALLOW_ON_SERVICE_FAILURE	True

Table 7-58 Bulwark Serivce Advanced Settings for Http Client and Request Timeout Configurations.

Key	Default Value
bulwark.service.enabled	true
BULWARK.REQUEST.URL	http://localhost:30900/v1/locks
BULWARK.REQUEST.TIMEOUT	3000
BULWARK.CONNECTION.HTTP2.ENABLED	true
BULWARK.CONNECTION.HTTP2.IDLE.TIMEOUT	600000
BULWARK.CONNECTION.HTTP2.CONNECT.TIMEOUT	3000
BULWARK.CONNECTION.HTTP2.MAX.CONNECTIONS.PER.DESTINATION	6
BULWARK.CONNECTION.HTTP2.MAX.REQUESTS.QUEUED.PER.DESTINATION	5000
BULWARK.CONNECTION.HTTP2.INITIAL.SESSION.RECV.WINDOW	16777216

Table 7-59 PDS Service Advanced Settings for Http Client and Request Timeout Configurations.

Key	Default Value
POLICYDS.REQUEST.URL	http://localhost:8084/pds/v2/user-data
POLICYDS.REQUEST.TIMEOUT	3000
POLICYDS.CONNECTION.HTTP2.ENABLED	false
POLICYDS.CONNECTION.HTTP2.IDLE.TIMEOUT	600000
POLICYDS.CONNECTION.HTTP2.CONNECT.TIMEOUT	3000
POLICYDS.CONNECTION.HTTP2.MAX.CONNECTIONS.PER.DESTINATION	6
POLICYDS.CONNECTION.HTTP2.MAX.REQUESTS.QUEUED.PER.DESTINATION	5000
POLICYDS.CONNECTION.HTTP2.INITIAL.SESSION.RECV.WINDOW	16777216

Table 7-60 PRE Service Advanced Settings for Http Client and Request Timeout Configurations.

Key	Default Value
Policy.PREEngineEnabled	true
PRE.REQUEST.URL	http://localhost:5806/v1/policy/engine/pcrf-core
PRE.REQUEST.TIMEOUT	2000
PRE.CONNECTION.HTTP2.ENABLED	false
PRE.CONNECTION.HTTP11.IDLE.TIMEOUT	600000
PRE.CONNECTION.HTTP11.CONNECT.TIMEOUT	3000
PRE.CONNECTION.HTTP11.MAX.CONNECTIONS.PER.DESTINATION	6
PRE.CONNECTION.HTTP11.MAX.REQUESTS.QUEUED.PER.DESTINATION	5000
PRE.CONNECTION.HTTP11.INITIAL.SESSION.RECV.WINDOW	16777216

Table 7-61 Binding Servicew Advanced Settings for Http Client and Request Timeout Configurations.

Key	Default Value
BindingService.Enabled	true
BINDING.REQUEST.URL	http://localhost:8080/binding/v1
BINDING.REQUEST.TIMEOUT	3000
BINDING.CONNECTION.HTTP2.ENABLED	false
BINDING.CONNECTION.HTTP2.IDLE.TIMEOUT	600000
BINDING.CONNECTION.HTTP2.CONNECT.TIMEOUT	3000
BINDING.CONNECTION.HTTP2.MAX.CONNECTIONS.PER.DESTINATION	6
BINDING.CONNECTION.HTTP2.MAX.REQUESTS.QUEUED.PER.DESTINATION	5000
BINDING.CONNECTION.HTTP2.INITIAL.SESSION.RECV.WINDOW	16777216

Table 7-62 Usage Monitoring Servicew Advanced Settings for Http Client and Request Timeout Configurations.

Key	Default Value
USAGEMON.REQUEST.URL	http://localhost:1080/occnp-usage-mon/v1/um-sessions
USAGEMON.REQUEST.TIMEOUT	3000
USAGEMON.CONNECTION.HTTP2.ENABLED	true
USAGEMON.CONNECTION.HTTP2.IDLE.TIMEOUT	600000
USAGEMON.CONNECTION.HTTP2.CONNECT.TIMEOUT	3000

Table 7-62 (Cont.) Usage Monitoring Servicew Advanced Settings for Http Client and Request Timeout Configurations.

Key	Default Value
USAGEMON.CONNECTION.HTTP2.MAX.CONNECTIONS.PER.DESTINATION	6
USAGEMON.CONNECTION.HTTP2.MAX.REQUESTS.QUEUED.PER.DESTINATION	5000
USAGEMON.CONNECTION.HTTP2.INITIAL.SESSION.RECV.WINDOW	16777216

Table 7-63 Advanced Settings to exclude specific APNs for the User Data Types

Key	Description
USER.allDataTypes.excludeApns	List the Access Point Names (APNs) for which all the requests or sessions from the PDS requests will be excluded for all the User Data Types (even if they are enabled). The list can include the comma separated values. Regular expressions are allowed for the APN names with the following conditions: • must not contain white spaces. • must be case insensitive. • can not be an empty string. • '\' (backslash) must be specified as '/d'. • the names must be separated only with commas (,).
USER.smPolicyData.excludeApns	List the APNs for which all the requests or sessions from the PDS requests will be excluded for the given SmPolicyData (even if they are enabled).
USER.ldapData.excludeApns	List the APNs for which all the requests or sessions from the PDS requests will be excluded for the given Ldap Data (even if they are enabled).
USER.ocsSpendingLimit.excludeApns	List the APNs for which all the requests or sessions from the PDS requests will be excluded for the given OCS Spending Limit (even if they are enabled).
USER.operatorSpecificData.excludeApns	List the APNs for which all the requests or sessions from the PDS requests will be excluded for the given Operator Specific Data (even if they are enabled).
USER.ssv.excludeApns	List the APNs for which all the requests or sessions from the PDS requests will be excluded for SSV (even if they are enabled).

Table 7-64 Advanced Settings Configurations for handling the race condition between Gx and Sy over two sites for CCR-U.

Key	Default Value
DIAMETER.Gx.Update.RaceModeratorEnabled	This is used to enable or disable race condition detection on CCR-U. If this flag is enabled, on detecting the race condition the following settings are used to decide the actions to take: • DIAMETER.Gx.Update.RetryOnRarRaceEventTriggers • DIAMETER.Gx.Update.RetryAttemptsOnRarRace • DIAMETER.Gx.Update.RetryWaitTimeOnRarRace • DIAMETER.Gx.Update.RejectionErrorCodeOnRarRace **Supported values**: true/false **Default value**: false
DIAMETER.Gx.Update.RetryOnRarRaceEventTriggers	If `DIAMETER.Gx.Terminate.RaceModeratorEnabled` is true, then a validation is made for all of the EventTriggers in the CCR-U request that are also included in this setting. **Note**: There must be at least one `registeredHandler` present in the Gx session that will make use of this event. If the validation fails for at least one of them, then the CCR-U will be put on hold, read the session again, and retry based on these settings: • DIAMETER.Gx.Update.RetryAttemptsOnRarRace • DIAMETER.Gx.Update.RetryWaitTimeOnRarRace • DIAMETER.Gx.Update.RejectionErrorCodeOnRarRace **Supported values**: These values can be provided as comma separated list. **Rx Triggers**: • ACCESS_NETWORK_INFO_REPORT • SUCCESSFUL_RESOURCE_ALLOCATION • LOSS_OF_BEARER **Sd Triggers**: • APPLICATION_START • APPLICATION_STOP **Default value**: ACCESS_NETWORK_INFO_REPORT
DIAMETER.Gx.Update.RetryAttemptsOnRarRace	The maximum number of retries when `DIAMETER.Gx.Update.RetryOnRarRaceEventTriggers` validation fails. **Supported values**: integer (0-n) **Default value**: 1
DIAMETER.Gx.Update.RetryWaitTimeOnRarRace	The amount of delay in milliseconds between retries when `DIAMETER.Gx.Update.RetryOnRarRaceEventTriggers` validation fails. **Supported values**: integer (0-n) **Default value**: 100

Table 7-64 (Cont.) Advanced Settings Configurations for handling the race condition between Gx and Sy over two sites for CCR-U.

Key	Default Value
DIAMETER.Gx.Update.RejectionErrorCodeOnRarRace	The error code response for the CCA-U when `DIAMETER.Gx.Update.RetryOnRarRaceEventTriggers` validation fails, and `DIAMETER.Gx.Update.RetryAttemptsOnRarRace` retries are exhausted. **Supported values**: Diameter Error codes • 5012 (DIAMETER_UNABLE_TO_COMPLY) • 3002 (DIAMETER_UNABLE_TO_DELIVER) **Default value**: 5012 **Note**: If invalid value is provided, fallback will be the default value 5012.

Table 7-65 Advanced Settings Configurations for handling the race condition between Gx and Sy over two sites for CCR-T.

Key	Default Value
DIAMETER.Gx.Terminate.RaceModeratorEnabled	This is used to enable or disable race condition detection on CCR-T. If this flag is enabled, on detecting the race condition the following settings are used to decide the actions to take: • DIAMETER.Gx.Terminate.RetryOnRarRaceEventTriggers • DIAMETER.Gx.Terminate.RetryAttemptsOnRarRace • DIAMETER.Gx.Terminate.RetryWaitTimeOnRarRace • DIAMETER.Gx.Terminate.RejectionErrorCodeOnRarRace **Supported values**: true/false **Default value**: false
DIAMETER.Gx.Terminate.RetryOnRarRaceEventTriggers	If `DIAMETER.Gx.Terminate.RaceModeratorEnabled` is true, then a validation is made for all of the EventTriggers in the CCR-T request that are also included in this setting. **Note**: There must be at least one `registeredHandler` present in the Gx session that will make use of this event. If the validation fails for at least one of them, then the CCR-T will be put on hold, read the session again, and retry based on these settings: • DIAMETER.Gx.Terminate.RetryAttemptsOnRarRace • DIAMETER.Gx.Terminate.RetryWaitTimeOnRarRace • DIAMETER.Gx.Terminate.RejectionErrorCodeOnRarRace **Supported values in comma separated list**: **Rx Triggers**: • ACCESS_NETWORK_INFO_REPORT • SUCCESSFUL_RESOURCE_ALLOCATION • LOSS_OF_BEARER **Sd Triggers**: • APPLICATION_START • APPLICATION_STOP **Default value**: ACCESS_NETWORK_INFO_REPORT

Table 7-65 (Cont.) Advanced Settings Configurations for handling the race condition between Gx and Sy over two sites for CCR-T.

Key	Default Value
DIAMETER.Gx.Terminate.RetryAttemptsOnRarRace	The maximum number of retries when `DIAMETER.Gx.Terminate.RetryOnRarRaceEventTriggers` validation fails. **Supported values**: integer (0-n) **Default value**: 1
DIAMETER.Gx.Terminate.RetryWaitTimeOnRarRace	The amount of delay in milliseconds between retries when `DIAMETER.Gx.Terminate.RetryOnRarRaceEventTriggers` validation fails. **Supported values**: integer (0-n) **Default value**: 100
DIAMETER.Gx.Terminate.RejectionErrorCodeOnRarRace	The error code response for the CCA-T when `DIAMETER.Gx.Terminate.RetryOnRarRaceEventTriggers` validation fails, and `DIAMETER.Gx.Terminate.RetryAttemptsOnRarRace` retries are exhausted. **Supported values**: Diameter Error codes • 5012 (DIAMETER_UNABLE_TO_COMPLY) • 3002 (DIAMETER_UNABLE_TO_DELIVER) **Default value**: 5012 **Note**: If invalid value is provided, fallback will be the default value 5012.

9. (Optional) To avoid sending stale session notification to PGW for Gx sessions when maxTTL value is reached, user can add **Override Cleanup Audit** parameter as AVP with its value set to true. To do so, perform the following steps:

   a. Expand the **Advanced Settings** group and click ⊕ Add
   The page opens the **Add Advanced Settings** dialog box.

   b. On the **Add Advanced Settings** dialog box, enter the follwing **Keys** and respective values:

Table 7-66 Add Advanced Settings Key and Value

Key	Default Value	Description
DIAMETER.Cleanup.OverrideCleanupAudit	True	This parameter is used along with `maxTTL` flag. If true along with `maxTTL` is also true, deletes the Gx and Rx stale sessions and no notification is sent to PGW. If false, notification is sent to PGW and stale sessions are not deleted.
DIAMETER.ENF.DUPLICATE.UPDATE.MESSAGE.HANDLING	false	Specifies if the duplicate message handling feature should be enabled or disabled.
DIAMETER.MsgBufferThreadCount	60	Specifies the number of threads to process read-diameter-messages.

Table 7-66 (Cont.) Add Advanced Settings Key and Value

Key	Default Value	Description
DIAMETER.MsgBufferQueue Size	8192	Specifies the size of the queue for holding pending read-diameter-messages.
DIAMETER.AF.MaxAFReport Handler	The minimum valid value for this parameter is 1. The maximum valid value of this setting is defined according to the following criteria: • For an Rx session with 2 media components (Audio, video) and 2 media sub components for each media component, the maximum value allowed is 2. • For an Rx session with 1 media component (Audio) and 2 media sub components, the maximum value allowed is 4. **Note** To enable the feature flag, Enter the value as *true* in the **Value** field.	Specifies the number of the Rx session to establish. **Note**: These minimum and maximum values are only valid with data compression disabled.
DIAMETER.ENF.AFDirectRepl y	True **Note:** Allowed value True or False	If true, PCRF responds immediatly to AAR request with AAA answer to AF, and at the same time RAR request is sent to PGW. If false, PCRF awaits until RAA is received from PGW and then responds with AAA answer to AF.
DIAMETER.AF.MaxWaitForInf oOnTermination	3000 milliseconds **Note:** Allowed value 0 - 10000 milliseconds.	The PCRF maximum wait time in milliseconds to receive the information from PCEF before responding to the AF with STR request. On not receiving the awaited information within the configured waiting period, PCRF resoponds to AF with successful STA answer without the requested information.

Table 7-66 (Cont.) Add Advanced Settings Key and Value

Key	Default Value	Description
DIAMETER.AF.IgnorePartialCodec	false	It specifies wheather to ignore QoS calculation if the uplink and downlink codec values are received in separate Rx messages. For example, uplink codec data in AAR-I and downlink codec data in AAR-U.
POLICYDS_REQUEST_TIMEOUT	3000 milliseconds **Note:** Allowed value 0 - 10000 milliseconds.	The PCRF maximum wait time for receiving response from PDS. On not receiving the awaited information from PDS within the configured waiting period, PCRF times out.
DB.GX.DATA.ENCODING.Enabled	false	When the value of this key is set to `true`, PCRF Core gxsession data is encoded. The JSON parameters are encoded based on the version indicated by `DB.GX.ENCODING.MAP.Version` value. If `DB.GX.ENCODING.MAP.Version` is not configured, its default version will be "gx0" and the parameter is encoded by default mapping. The PCRF Core gxsession data is decoded based version value in the session.
DB.RX.DATA.ENCODING.Enabled	false	When the value of this key is set to `true`, PCRF Core rxsession data encoding is enabled. The JSON parameters are encoded based on the version indicated by `DB.RX.ENCODING.MAP.Version` value. If `DB.RX.ENCODING.MAP.Version` is not configured, its default version will be "rx0" and the parameter is encoded by default mapping. The PCRF Core rxsession data is decoded based version value in the session.

Table 7-66 (Cont.) Add Advanced Settings Key and Value

Key	Default Value	Description
DB.SD.DATA.ENCODING.Enabled	false	When the value of this key is set to `true`, PCRF Core sdsession data is encoded. The JSON parameters are encoded based on the version indicated by `DB.SD.ENCODING.MAP.Version` value. If `DB.SD.ENCODING.MAP.Version` is not configured, its default version will be "sd0" and the parameter is encoded by default mapping. The PCRF Core sdsession data is decoded based version value in the session.
DB.GX.ENCODING.MAP.Version	0	Indicates the mapping version name for gxsession encoding. To set this use a "unique string" such as "gx0". The encoding version is used for decoding. When value of this key is set to "0", or this key is not configured, the encoding/ decoding uses default map.
DB.RX.ENCODING.MAP.Version	0	Indicates the mapping version name for rxsession encoding. To set this use a "unique string" such as "rx0". The encoding version is used for decoding. When value of this key is set to "0", or this key is not configured, the encoding/ decoding uses default map.
DB.SD.ENCODING.MAP.Version	0	Indicates the mapping version name for sdsession encoding. To set this use a "unique string" such as "sd0". The encoding version is used for decoding. When value of this key is set to "0", or this key is not configured, the encoding/ decoding uses default map.

Table 7-66 (Cont.) Add Advanced Settings Key and Value

Key	Default Value	Description
DB.ENCODING.MAP.LIST	0	Indicates the mapping list keyMap and valueMap in JSON format. When this value is set to "0" the default map is referred for encoding and decoding. To set this to cusomized map in the form of JSON format containing keyMap and valueMap.

c. Click **Save**.
This saves the advanced settings and the details get listed on **Edit Settings** page under **Advanced Settings**.

10. Click **Save**.
The page saves the PCRF Core service settings.

7.3.8.2 Serving Gateway

This procedure provides information about how to create and manage serving gateways.

The **Serving Gateway** page allows you to create new and manage existing serving gateways. The page displays the list of defined serving gateways and provides the options to import, export, or add gateways.

To configure serving gateway:

1. From the navigation menu under **Policy**, navigate to **Service Configurations**, click **PCRF Core**, and then select **Serving Gateway**.
This opens the **Serving Gateway** page. The page lists the existing gateway details. You can add or import new serving gateways using this page.

> **Note**
>
> Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
This opens the **Create Service Gateway** page.
3. On the **Create Service Gateway** page, enter the following information:

Table 7-67 Create Service Gateway

Field Name	Description
Name	The customized name for serving gateway
Description	The customized description for serving gateway
MCC-MNC	The Mobile country code (3-digit number) or Mobile network code (2- or 3-digit number)
Serving Gateway IP Address/Subnet	IP address or subnet for the serving gateway

4. Click **Save**.
   The serving gateway gets listed on the **Serving Gateway** page.

> **Note**
>
> Use ✎ or 🗑 available under the **Actions** column to update or delete the serving gateway.

Importing Serving Gateway

To import Serving Gateway:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.3.8.3 Network Element

This section describes how to configure network elements.

To access network elements configurations from CNC Console home page, under **Policy**, navigate to **Service Configurations**, click **PCRF Core**, and then select **Network Elements**.

On clicking PCRF Core, you can select to customize any of the following network elements:

- PGW (PDN Gateway)
- GGSN (GPRS Support Node)

7.3.8.3.1 PGW

This procedure provides information about how to create and manage PDN Gateway (PGW) network element.

The **PGW** page allows you to create new and manage existing PGWs. The page displays the list of defined PGWs with the options to import, export, or add profiles.

To configure PGW:

1. From the navigation menu under **Network Elements**, select **PGW**.
   This opens the **PGW** page. The page lists the existing PGWs. You can add or import new PGWs using this page.

> **Note**
>
> Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create PGW** page.
3. On the **Create PGW** page, enter the following information:

Table 7-68 Create PGW

Field Name	Description
Name	Use this field to add a customized name for PGW.
Description/Location	Use this field to add a customized description or location for network element.
Backup Host Name/IP Address	Registered domain name, or IP address in IPv4 or IPv6 format, assigned to the network element.
Host Name/IP Address	Alternate address that is used if communication between the CNC Policy and the primary address for the network element fails.
Capability	Specifies the capability for the network element. PGW network elements are compatible with usage_report event trigger value 26.
Capacity	Specifies the bandwidth allocated to the network element.
IP Domain ID	Specifies the IPv4 domain identity. This value uniquely identifies the network element if the same IPv4 address is assigned in multiple networks.
Diameter Realm	Specifies the realm to be sent in the diameter message.
MIP6 Host Identity	Specifies the Mobile IPv6 (MIPv6) host.
Diameter Identity	Specifies the fully qualified domain name (FQDN) of the network element

4. Click **Save**.
   The PGW gets listed on the **PGW** page.

> **Note**
>
> Use  or  available under the **Actions** column to update or delete the PGW.

Importing PGW

To import PGW:

1. Click  .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.3.8.3.2 GGSN

This procedure provides information about how to create and manage PDN Gateway (GGSN) network element.

The **GGSN** page allows you to create new and manage existing GGSNs. The page displays the list of defined GGSNs with the options to import, export, or add profiles.

To configure GGSN:

1. From the navigation menu under **Network Elements**, select **GGSN**.

This opens the **GGSN** page. The page lists the existing GGSNs. You can add or import new GGSNs using this page.

> **Note**
>
> Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
This opens the **Create GGSN** page.
3. On the **Create GGSN** page, enter the following information:

Table 7-69 Create GGSN

Field Name	Description
Name	Use this field to add a customized name for GGSN
Description/Location	Use this field to add a customized description or location for network element
Backup Host Name/IP Address	Registered domain name, or IP address in IPv4 or IPv6 format, assigned to the network element
Host Name/IP Address	Alternate address that is used if communication between the CNC Policy and the primary address for the network element fails
Capability	Specifies the capability for the network element. Usually, GGSN network elements are compatable with usage_report event trigger value 26
Capacity	Specifies the bandwidth allocated to the network element
IP Domain ID	Specifies the IPv4 domain identity. This value uniquely identifies the network element if the same IPv4 address is assigned in multiple networks
Diameter Realm	Specifies the realm to be sent in the diameter message
Diameter Identity	Specifies the fully qualified domain name (FQDN) of the network element

4. Click **Save**.
The GGSN gets listed on the **GGSN** page.

> **Note**
>
> Use or available under the **Actions** column to update or delete the GGSN.

Importing GGSN

To import GGSN:

1. Click Import .

The page opens the **File Upload** dialog box.

2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.3.9 Audit Service

This procedure provides information about configuring the Audit Service.

The **Audit Service** page displays the Audit Service configurations. The page allows you to edit the configurations.

To configure Audit service:

1. From the navigation menu under **Policy**, navigate to **Service Configurations**, and select **Audit Service**.
   This opens the **Audit Service** page. The page displays the existing configurations.
2. Click 

   .
   This opens the **Edit Audit Service** page.
3. Make sure that the value of the **Audit Enabled** switch is enabled.
   This field determines if auditing must be enabled for all the registered Policy services. By default, this switch remains disabled.
4. Expand the **Forced Deletion** group and configure the **Minimum Audit Attempts** parameter.

   Minimum Audit Attempts specifies the minimum number of audit attempts until ForceTTL is reached.

   If ForceTTL is reached and audit_attempts >= Minimum Audit Attempts of ForceTTL, then Audit service deletes the identified stale records from its respective database.

   For example, it deletes the stale SM sessions from SMPolicyAssociation table.

   The default value of this parameter is 0 and the value can range between 0 to 255.
5. Click **Save**.
   The page saves the Audit service configurations.

Note

- Perform the Audit configurations at the individual service level (SM service and Binding service), for Audit service and notifications to work appropriately.

  Important considerations during Audit configurations:

  In situations where the number of stale records (records that are eligible to be audited) at any given point of time in the system (for a given microservice database) is expected to be high (for example, > 1M), then the **Policy Session Age** and/or the **Notification Rate** parameters should be set appropriately such that at least 2 audit cycles can be finished before the records that were assessed by the first cycle fall stale again. The Audit procedure having a **Policy Association Age** less than this recommendation, may not be able to assess all stale records as the already assessed ones will be stale again too soon. It is recommended to keep a minimum of 24 hours for the **Policy Association Age** and a minimum of 48 hours for **Max Policy Association Age**.

  The time taken to complete an Audit Cycle and begin the next one can be calculated as below:

  Audit Cycle Time = S / (N*60) + I minutes, where,

  S = expected number of stale sessions at any given time,

  N = notification rate (per second),

  I = minimum Audit Interval

- When table slicing feature is enabled, and audit is running on a set of sliced tables, the audit notification rate would be expected to reach no of sliced tables * configured notification rate if audit is running for some/all the sliced tables concurrently.

Audit Schedule Data

Audit service supports multiple pod, using the Audit Schedule and Audit TaskScheduler. Audit schedule has list of registerd services that needs audit service with all the scheduled details. Audit TaskScheduler polls for all those audit jobs in the QUEUED status.

Figure 7-14 Audit Scheduled Service List Data

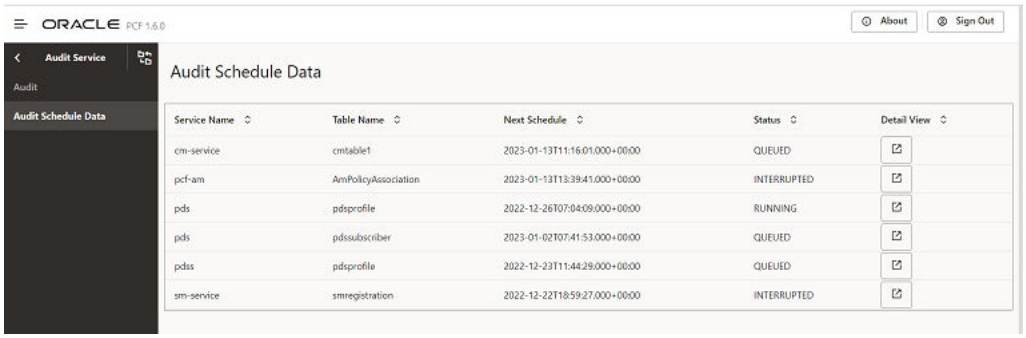


Table 7-70 Audit Schedule Data Fields

Field Name	Description
Service Name	Name of the service that had registerd for audit service.
Table Name	Name of the database table that has requested for the audit service
Next Schedule	Next schedule availability time
Status	Scheduler job status
Start Time	Audit start time of pod.
End Time	Audit end time of pod.
Last Polled Time	Pod and associated scheduler details.
Schedular Details	Last time at which pod updated this table.
Is Service Dependent	If this is true, notification will be triggered for one table in the service, for all table's in the service, notifications will be sent parallelly based on the notificationRate set for each table.
Priority	Priority set by customer to audit certain records or group of records in some order
Version	For JPA optimization

Audit Schedule task can be in one of the state described below:

Figure 7-15 Audit Schedule Job Work Flow

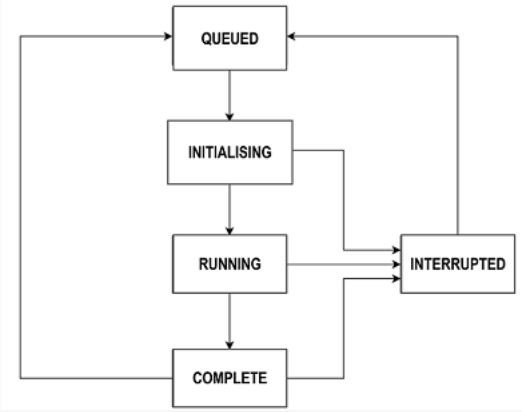


Table 7-71 Audit Schedule Job Status

Status	Description
QUEUED	When a registered request is received it is added to the schedule table, the job status is set to Queued.
INITIALISING	When polling task is complete and is initialising for AuditTaskManager to audit, the job status is set to Initialising.
RUNNING	When Audit task manager starts the task, the job status is set to Running.

Table 7-71 (Cont.) Audit Schedule Job Status

Status	Description
COMPLETE	When Audit TaskManager completes the Audit, the status is set to Complete.
INTERRUPTED	When the audit process is paused from GUI, then the job status is set to Interrupted. When resumed from GUI, job status is set to Queued When deregistered, the job data will be removed from the table. when a service is register, it is set to Queued.
DEREGISTERED	When deregistered, the job status is set to Deregistered for Initialising or Running and the entry will be deleted for any other status(except Initialising or Running). In case the a service register request is received before AuditTaskManager deletes the table entry for deregistered job statuses, the status would be moved to Queued.

For more information on Audit Service and Audit Schedule REST API details, see the section *Audit Service* in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

7.3.10 Configuring Notifier Service

This section provides information on how to configure Notifier service.

7.3.10.1 Notifier Configurations

This procedure provides information about managing Notifier Service Settings.

The Notifier Service Settings page allows you to edit Advanced Settings for Notifier service.

To configure, perform the following steps:

1. From the navigation menu, under **Policy**, click **Service Configurations**, click **Notifier**, and then select **Notifier Configurations**.
   This opens the Notifier Service Settings page.
2. Click **Edit**.

   This opens the Edit Notifier Service Settings page.
3. Edit the **SMPP Configuration**:

Table 7-72 SMPP Configuration

Attribute Name	Description
Enable SMPP	Enables the Subscriber Notification using SMPP Protocol. **Default Value**: false

Table 7-72 (Cont.) SMPP Configuration

Attribute Name	Description
Default SMS Gateway Group	All the SMS Gateways that are created in SMS GATEWAY CONFIGURATION page.
ESME Source Address	External Short Messaging Entity (ESME) Source Address. By default this field is empty.
ESME Source Address TON	Indicates the ESME Source address type of number: • UNKNOWN • INTERNATIONAL • NATIONAL • NETWORK_SPECIFIC • SUBSCRIBER_NUMBER • ALPHANUMERIC • ABBREVIATED **Default Value**: UNKNOWN
ESME Source Address NPI	ESME Source Address Number Plan Indicator: • UNKNOWN • ISDN (E163/E164) • DATA (X, 121) • TELEX • LAND_MOBILE • NATIONAL • PRIVATE • ERMES • INTERNET • WAP **Default Value**: UNKNOWN
Connection Timeout (in milliseconds)	Indicates the SMS Gateway Connection Timeout in milliseconds. **Range**: 1-60000 **Default Value**: 30000
Response Timeout (in milliseconds)	Indicates the timeout (in milliseconds) for the response going towards SMS Gateway. **Range**: 1-60000 **Default Value**: 10000

4. Click **Add** to add the advanced settings.

Table 7-73 Advanced Settings

Key	Value
smpp.bindType	Bind type of the SMPP server. Allowed values: • BIND_TX • BIND_RX • BIND_TRX **Default value**: BIND_TRX

Table 7-73 (Cont.) Advanced Settings

Key	Value
smpp.systemType	System type. This key can accept null values.
smpp.addrTon	Address type of number of the SMPP server. Allowed values: • UNKNOWN • INTERNATIONAL • NATIONAL • NETWORK_SPECIFIC • SUBSCRIBER_NUMBER • ALPHANUMERIC • ABBREVIATED **Default value**: UNKNOWN
smpp.addrNpi	Address Number Plan Indicator: • UNKNOWN • ISDN (E163/E164) • DATA (X, 121) • TELEX • LAND_MOBILE • NATIONAL • PRIVATE • ERMES • INTERNET • WAP **Default value**: UNKNOWN
smpp.addrRange	Set of SME addresses. Address range can accept null values.
smpp.serviceType	Service type can be either null or CMT.
smpp.encodingScheme	SMPP Encoding Scheme. **Default value**: UTF-8
smpp.destAddrTon	Destination address type of number: • UNKNOWN • INTERNATIONAL • NATIONAL • NETWORK_SPECIFIC • SUBSCRIBER_NUMBER • ALPHANUMERIC • ABBREVIATED **Default value**: UNKNOWN

Table 7-73 (Cont.) Advanced Settings

Key	Value
smpp.destAddrNpi	Destination Address Number Plan Indicator: • UNKNOWN • ISDN (E163/E164) • DATA (X, 121) • TELEX • LAND_MOBILE • NATIONAL • PRIVATE • ERMES • INTERNET • WAP **Default value**: ISDN
smpp.validatePeriodInDays	Validity period of the SMS (in days) **Default value**: 3 days
smpp.esmClassValue	The message mode and type, indicating any special message attributes associated with the SMS. **Default value**: 0
smpp.reconnectMinDelayMillis	Minimum delay in milliseconds after the consumer/producer tries to reconnect to the SMSC, after the connection was lost. **Default value**: 500
smpp.reconnectMaxDelayMillis	Maximum delay in milliseconds after the consumer/producer tries to reconnect to the SMSC, after the connection was lost. **Default value**: 10000
smpp.smDefaultMsgId	Short message default message identifier. **Default value**: 0
smpp.protocolId	Protocol ID. **Default value**: 0
smpp.priorityFlag	Priority Flag. **Default value**: 0
smpp.packedGSM7Encoding	Indicates whether to apply GSM-7 character encoding. **Default value**: false
smpp.sarSegmentSize	SMS Segment size. **Default value**: 134

5. Click **Save** to save the advanced settings.
   To discard the changes, click **Cancel**
6. Click **Save** to save the Notifier Service Settings.
   To discard the changes, click **Cancel**

7.3.10.2 Notification Server

7.3.10.2.1 HTTP Server

This procedure provides information on how to add and manage HTTP Server.

The HTTP Server page allows you to add and edit HTTP servers.

To configure, perform the following steps:

1. From the navigation menu, under **Policy**, click **Service Configurations**, click **Notifier**, click **Notification Server**, and then select **HTTP Server**.
   This opens the HTTP Server page.
2. Click **Add**.

   This opens the Add HTTP Server page.
3. Enter values for the following input fields, as described in the following table:

Table 7-74 Add HTTP Server

Field Name	Description
Name	Enter the name of the HTTP server. The name can only contain the characters A through Z, a through z, 0 through 9, period (.), hyphen (-), and underline (_).
Description/Location	Free-form text that identifies the HTTP server within the network. Enter up to 250 characters.
Scheme	Select the scheme. The available values are: • HTTP • HTTPS
Host	Enter the Host Name of the HTTP server.
Alt Host	Enter the Alternate Host Name of the HTTP server.
URI Path	Enter the URI path.
HTTP Version	Select the HTTP version. The available values are: • 1.1 • 2
HTTP Header	
Header Key	Enter the header key.
Header Value	Enter the value for the header key.

4. Click **Save**. The HTTP server is displayed on the HTTP Server page.
   To discard the changes, click **Cancel**

Use pencil icon or trash bin icon available in the next column to edit or update the created HTTP server.

7.3.10.2.2 SMS Gateway Group

The SMS Gateway Group page allows you to configure the SME Gateways.

1. From the navigation menu, under **Policy**, click **Service Configurations**, click **Notifier**, click **Notification Server**, and then select **SMS Gateway Group**.
   This opens the SMS Gateway Group page.
2. Click **Add**.

   This opens the Add SMS Gateway Group page.
3. Enter values for the following input fields, as described in the following table:

Table 7-75 Add SMS Gateway Group

Field Name	Description
Name	Enter the name of the SMS Gateway server. The name can be Maximum 40 characters consisting of alphabets, numbers, period, hyphen and underscore (_). By default, the value is left empty.
Description	Free-form text that identifies the SMS Gateway server within the network. Enter up to 250 characters. By default, the value is left empty.
Primary SMSC Host	Enter the Host Name of the primary SMSC Host server. By default, the value is left empty.
Request Delivery Receipt	Specify the mode of delivery receipt. This field accepts the following integer values: • 0 to indicate REQUEST_DELIVERY_RECEIPT_NO • 1 to indicate REQUEST_DELIVERY_RECEIPT_SUCCESS_FAILURE • 2 to indicate REQUEST_DELIVERY_RECEIPT_FAILURE **Default Value**: 0
Maximum Byte Length	Maximum number of bytes allowed for the message. The length of the message can range between 1 to 254 bytes. **Default Value**: 140
Long Message Enabled	Enables the long messages. **Default Value**: true
Long Message maximum Byte Length	Specifies the maximum number of bytes allowed in the long message. Maximum length of the long messages can range between 1 to 999 bytes. **Default Value**: 500 For details on the long message scenarios, see Long Message Scenarios
Long Message Delivery Method	Specify the long message delivery method. It can take the following integer values: • 0 to indicate SEGMENTATION AND REASSEMBLY (SAR) to deliver the long messages in segments • 1 to indicate the MESSAGE_PAYLOAD to deliver the long messages in payload **Default Value**: 0
Truncate Long Messages	Indicates whether to truncate messages longer than the specified length. **Default Value**: false

Table 7-75 (Cont.) Add SMS Gateway Group

Field Name	Description
Fail Max Message Size	Specify the maximum length of the message after which the message fails. **Default Value**: true

Table 7-76 Long Message Scenarios

Scenario	Fail Max Message Size	Long Message Enabled	Truncate Long Messages	Result
Message with 165 charactersLength > 140 and < 254	Enabled	Disabled	Disabled	• Message failed with cause: SmsLengthExceeded • "message":"rejecting message because longMsgEnabled: false, messageLen: 165 > maxBytesLen: 140, failMsgOnMaxMsgSize: true",
Message with 165 charactersLength > 140 and < 254	Disabled	Disabled	Disabled	Message has been successfully sent to the SMSC server without any error.
Message with 260 characters Length > 140 and > 254	Disabled	Disabled	Disabled	• Messaged failed with cause: SmsLengthExceeded • "message":"rejecting message because longMsgEnabled: false, messageLen: 140 > 254, truncateOnLongMsg: false"
Message with 165 characters Length > 140 and < 254	Disabled	Disabled	Enabled	Message has been truncated to 140 characters and sent to the SMSC server without any error.

Table 7-76 (Cont.) Long Message Scenarios

Scenario	Fail Max Message Size	Long Message Enabled	Truncate Long Messages	Result
Message with 260 charactersLength > 140 and > 254	Disabled	Disabled	Enabled	Message has been truncated to 140 characters and sent to the SMSC server without any error.

4. Click **Save**. The SMS Gateway details are displayed on the **SMS Gateway Group** page. To discard the changes, click **Cancel**
5. You can use pencil icon or trash bin icon available in the next column to edit or delete the created SMS Gateway server.

7.3.10.2.3 SMSC Host Info

The SMSC Host Info page allows you to add and edit SMSC host configuration.

To configure, perform the following steps:

1. From the navigation menu, under **Policy**, click **Service Configurations**, click **Notifier**, click **Notification Server**, and then select **SMSC Host Info**.
   This opens the SMSC Host Info page.
2. Click **Add**.

   This opens the Add SMSC Host Info page.
3. Enter values for the following input fields, as described in the following table:

Table 7-77 Add SMSC Host Info

Field Name	Description
Name	Enter the name of the SMSC Host server. The name can be lower case alphanumeric characters, '-' or '.', and must start and end with an alphanumeric character. By default, the value is left empty.
SMSC Host	Enter the Host Name of the SMSC server. The host can be Ipv4, Ipv6 or FQDN with optional port value. By default, the value is left empty.
SMSC Port	Enter the Port number of the SMSC server. Range: 0-65535 By default, the value is left empty.

4. Click **Save**. The SMSC Host server is displayed on the SMSC Host Info page.
   To discard the changes, click **Cancel**

Use pencil icon or trash bin icon available in the next column to edit or delete the SMSC Host details.

7.3.11 PDS

This section includes the Policy Data Service (PDS) configurations.

To access PDS configurations from the CNC Console home page, expand **Policy**, navigate to **Service Configurations** and select **PDS**.

On clicking **PDS**, you can customize the following configurations:

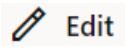
- PDS Settings
- PDS Workflow

7.3.11.1 PDS Settings

This procedure provides information about configuring the PDS settings.

The **PDS Settings** page under **PDS** displays the PDS settings. The page allows you to edit the configurations.

To edit the PDS settings:

1. From the navigation menu under **Policy**, navigate to **Service Configurations**, click **PDS**, and then select **PDS Settings**.
   This opens the **PDS Settings** page. The page displays the existing settings for PDS.
2. Click 

   .
   This opens the **Edit PDS Settings** page.
3. Expand the **Remote Subscriber State Variable Root Path** group.
   Remote Subscriber State Variable Root Path defines the root path of the remote subscriber state variables block corresponding to each datasources. For example, SMPolicyData, AMPolicyData, OperatorSpecificData, UEPolicyData, SpendingLimitData, and so on.

   For information about subscriber state variables block, see *Oracle Communications Cloud Native Core, Converged Policy Design Guide*. This information is written back (PATCH Request) to UDR as soon as the values are updated or when the last session of the subscriber terminates.
4. Under the **Remote Subscriber State Variable Root Path** group, enter values for the following parameters:

Table 7-78 Remote Subscriber State Variable Root Path Group Settings

Field Name	Description
SM Policy Data	Defines the path of subscriber state variables under SMPolicyData. You can provide multiple paths separated by commas.
UE Policy Set	Defines the path of subscriber state variables under UEPolicyData. You can provide multiple paths separated by commas.
Enable End Session SSV Remote to UDR	Determines if the dynamic write should happen at time of policy execution or at the end of the last session for subscriber.

Figure 7-16 PDS Settings: Remote Subscriber State Variable Root Path

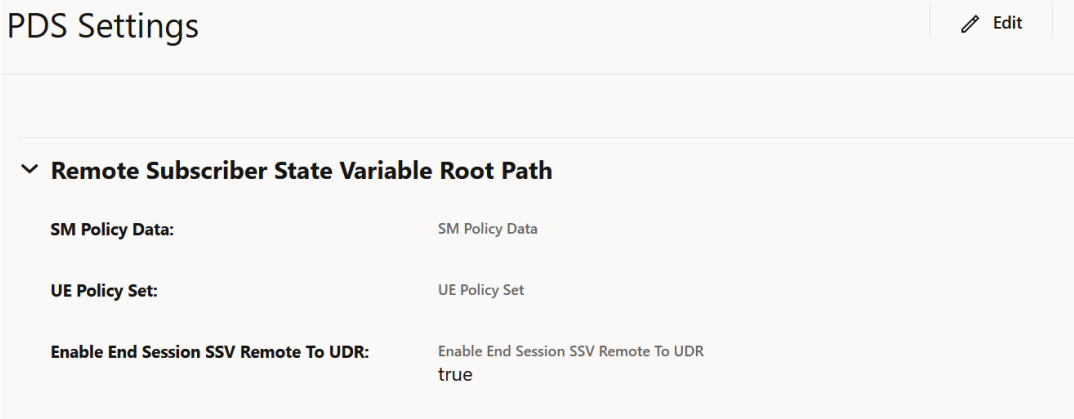
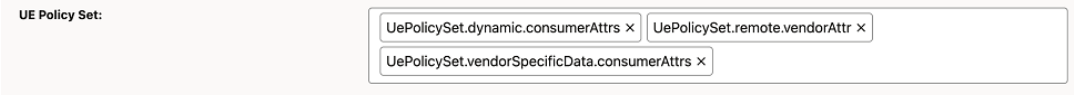


Figure 7-17 Sample UE Policy Set Configured



5. Expand the **Common** group and enter values for the following parameters:

Table 7-79 Common Group Settings

Field Name	Description
Enable HTTP2	If this switch is enabled, the HTTP2 remains enabled. By default, the switch remains disabled.
Request Timeout	If PDS Setting is updated then set the timeout as 3000 for the initial flow to succeed.
Type of Search	Specifies a key preference for PDS search query used to look up for a Pdsprofile record. The value can be **Multi-UE ID Search** or **Preferential Search** . When **Multi-UE ID Search** is selected the search query uses OR criteria and the search time will be higher. When **Preferential Search** is selected, Preference Order field is displayed. By default, the value for this parameter is set to **Multi-UE ID Search**.

Table 7-79 (Cont.) Common Group Settings

Field Name	Description
Preference Order	Specifies a list of PDS search indexes that user can select in any order of their preference. From drop-down menu, user can select from: **SUPI, GPSI, GPSI/SUPI, SUPI/GPSI**. If the PDS search happens using the search key **SUPI**, it will look up for Pdsprofile record based on SUPI. In case the request does not have SUPI, the request will be rejected. If the PDS search happens using the search key **SUPI/GPSI**, it will look up for Pdsprofile record based on SUPI. In case the request does not have SUPI, it will use GPSI as the search index. **Note**: If Preferential Search is selected, but Preference Order is not set, then default preference order will be: **GPSI,SUPI**.
Enable Fetch and Resubscribe	If this switch is set to true, PDS refetches the data and revalidates the subscription when the number of sessions exceeds the configured limit at PDS. PDS revalidates each subsequent request received after the configured limit. If the revalidation is successful, PDS does not perform any further action. However, if PDS fails to revalidate subscriptions (with error status 404), then it resubscribes with the datasource. By default, the value for this parameter is set to false. For `smPolicyData`, the limit for the number of sessions is defined per SNSSAI:DNN. **Note**: For user data type `smPolicyData`, the attributes **Include Snssai in User Query** and **Include Dnn in User Query** on the PCF Session Management page must be set to false. For `spendingLimitData`, the limit is defined per DNN:SNSSAI.
Data Compression Scheme	Specifies the data compression scheme used by PDS. Following are the allowed values for this field: • Disabled • Zlib_Compressed By default, this value remains disabled.

When the **Enable Fetch and Resubscribe** switch is set to true, you are required to configure the parameters described in this table.

Table 7-80 Session Count per DNN/APN List Settings

Field Name	Description
DNN	Displays the Data Network Name. The allowed expression for DNN is [^[A-Za-z0-9-.]+$] and NA. **Note**: Do not add spaces to this field.

Table 7-80 (Cont.) Session Count per DNN/APN List Settings

Field Name	Description
SST	Displays the Slice or Service Type name in the S-NSSAI. You can enter a number from 0 to 255 or NA.
SD	Displays the Slice Differentiator Name. The allowed expression for SD is ^([A-Fa-f0-9]+|NA)$. **Note**: Do not add spaces or any other characters to this field.
Max Sessions Count	Displays the number of sessions after which PDS refetches and revalidates subscriptions. You can enter a number from 1 to 20.

6. Under the **Audit** group, enter values for the following parameters:

Table 7-81 Configurable Parameters for Audit group

Field Name	Description
Enabled	Determines whether to send registration request to Audit service or not. **Default Value**: false
Notification Rate (per second)	Defines the number of stale records which Audit service notifies to PDS service in one second. **Default and Recommended Value**: 50 **Note**: To configure higher number than the recommended value, contact My Oracle Support (https://support.oracle.com)
Subscriber Session Age (in minutes)	Defines the age of PDS subscriber entry after which a record is considered to be stale on Policy and services (SM, AM, UE, and PCRF Core) are queried for presence of such associations. **Default Value**: 1500
Subscriber Session Max Age (in minutes)	Defines the age of PDS subscriber entry after which a record is purged from the database and no query requests are sent to services for presence of such associations. **Default Value**: 2940
Suscriber Min Audit Pass Interval (in minutes)	Defines the minimum interval between start times of two consecutive audit passes for the pdssubscriber table. **Default Value**: 390
PDS Profile Age (in minutes)	Defines the age of PDS profile (GPSI-SUPI-NAI mapping) after which the profile is considered to be stale in Policy and PDS subscriber table is queried for presence of such associations. **Default Value**: 4500
PDS Profile Max Age (in minutes)	Defines the age of PDS profile (GPSI-SUPI-NAI mapping) after which the profile is purged from the database without sending any queries to PDS subscriber table. **Default Value**: 8820

Table 7-81 (Cont.) Configurable Parameters for Audit group

Field Name	Description
PDS Profile Min Audit Pass Interval (in minutes)	Defines the minimum interval between start times of two consecutive audit passes for the pdssubscriber table. **Default Value**: 1170

> **Note**
>
> At the time of fresh installation, check and save PDS settings configuration even if there is no change to the configuration.
>
> a. Click **Edit**.
> b. Update the values if required.
> c. Click **Save** (even if there is no change to the configuration).
>
> The Audit Service starts auditing with the default/updated configuration.

7. Perform the following steps to configure **Advanced Settings** that allows PDS to handle concurrent requests:

   a. Click the ⊕ Add
   The page opens the **Add Advanced Settings** dialog box.
   b. In the dialog box, enter the following **keys** and respective **values**:

Table 7-82 Parameters for Advanced Settings

Keys	Value
NOTIFICATION_DATA_COMPARE_FLAG	When enabled, this flag compares the incoming notification content with existing data in the database for a subscriber. If there is any change identified, PDS notifies the context owner, otherwise notification will not be forwarded to core service (context-owner). **Default Value**: `false`
CONCURRENT_REQUEST_GUARD_TIME	On receiving `resetContext` flag for a resource from core-service, PolicyDS keeps the association per subscriber resource that are within the time period (in miliseconds). Any other associations which are beyond this time period are cleaned up. **Default Value**: 10000 milisec **Note**: When Usage Monitoring feature is enabled and Concurrency Control is enabled in Usage Monitoring settings, set the value of CONCURRENT_REQUEST_GUARD_TIME=0.
OPTIMISTIC_RETRY_MAX_ATTEMPT	In case of concurrent DB transactions, if DB operation failed then for these many number of times the DB operation can be re-attempted. **Default Value**: 2

Table 7-82 (Cont.) Parameters for Advanced Settings

Keys	Value
ACTION_ON_SSV_REMOTE_IN_REVALIDATION_OR_NOTIFICATION	Used to either override or ignore SSV during: • revalidation of subscription for N+ sessions • UDR notification **Default Value**: OVERRIDE
PDS_SEARCH_USING_All_UEIDS	Used when **typeOfSearch** field is set to **Preferential Search**. When set to true, the preference order will be ignored. PDS will use all the UE Ids available in the request as search indexes and use them one by one to search for Pdsprofile record. **Default Value**: false **Note**: When typeOfSearch field is set to **Multi-UE Id search** this configuration is not considered.
CONCURRENCY.BULWARK_ENABLED_FOR_CHF_NOTIFICATION	Determines whether to enable Bulwark service for CHF notification flow. **Default Value**: false
CONCURRENCY.LOCK_REQUEST_RETRY_COUNT_FOR_CHF_NOTIFICATION	The number of retries when a request to Bulwark service fails for CHF notification flow. **Default Value**: 3
CONCURRENCY.LOCK_LEASE_DURATION_FOR_CHF_NOTIFICATION	Duration in (milliseconds) for which the lock can be held for CHF Notification. **Default Value**: 2000 (ms)
CONCURRENCY.LOCK_WAIT_DURATION_FOR_CHF_NOTIFICATION	Duration in (milliseconds) for which the CHF can wait to acquire a lock once it sends a request to Bulwark service. **Default Value**: 3000 (ms)
CONCURRENCY.LOCK_REQUEST_KEY_TYPE_FOR_CHF_NOTIFICATION	Determines the key value that is used to aquire the lock for CHF Notification. **Default Value**: SUPI
CONCURRENCY.BULWARK_ENABLED_FOR_OCS_NOTIFICATION	Determines whether to enable Bulwark service for OCS notification flow. **Default Value**: false
CONCURRENCY.LOCK_REQUEST_RETRY_COUNT_FOR_OCS_NOTIFICATION	The number of retries when a request to Bulwark service fails for OCS notification flow. **Default Value**: 3
CONCURRENCY.LOCK_LEASE_DURATION_FOR_OCS_NOTIFICATION	Duration in (milliseconds) for which the lock can be held for OCS Notification. **Default Value**: 2000 (ms)
CONCURRENCY.LOCK_WAIT_DURATION_FOR_OCS_NOTIFICATION	Duration in (milliseconds) for which the OCS can wait to acquire a lock once it sends a request to Bulwark service. **Default Value**: 3000 (ms)
CONCURRENCY.LOCK_REQUEST_KEY_TYPE_FOR_OCS_NOTIFICATION	Determines the key value that is used to aquire the lock for OCS Notification. **Default Value**: SUPI

Table 7-82 (Cont.) Parameters for Advanced Settings

Keys	Value
CONCURRENCY.BULWARK_ENABLED_FOR_UDR_NOTIFICATION	Determines whether to enable Bulwark service for UDR notification flow. **Default Value**: false
CONCURRENCY.LOCK_REQUEST_RETRY_COUNT_FOR_UDR_NOTIFICATION	The number of retries when a request to Bulwark service fails for UDR notification flow. **Default Value**: 3
CONCURRENCY.LOCK_LEASE_DURATION_FOR_UDR_NOTIFICATION	Duration in (milliseconds) for which the lock can be held for UDR Notification. **Default Value**: 2000 (ms)
CONCURRENCY.LOCK_WAIT_DURATION_FOR_UDR_NOTIFICATION	Duration in (milliseconds) for which the UDR can wait to acquire a lock once it sends a request to Bulwark service. **Default Value**: 3000 (ms)
CONCURRENCY.LOCK_REQUEST_KEY_TYPE_FOR_UDR_NOTIFICATION	Determines the key value that is used to aquire the lock for UDR Notification. **Default Value**: SUPI
CONCURRENCY.LOCK_REQUEST_RETRY_COUNT	The number of retries when a request to Bulwark service fails. **Default Value**: 3
CONCURRENCY.LOCK_LEASE_DURATION	Duration for which lock is held once it is acquired. After this duration, the lock will be released automatically. **Default Value**: 6000 (ms)
CONCURRENCY.LOCK_WAIT_DURATION	Duration for which the Policy service will wait for the response to get a lock. The same duration is used by Bulwark service to poll for the lock if a lock is not available. **Default Value**: 3000 (ms)
BULWARK_ERROR_CODES	Error codes from Bulwark Service for which Notification can be rejected by PDS.
ENABLE_CUSTOM_ATTRIBUTES_IN_UDR_NOTIFICATION	Determines whether to disable customAttributes in UDR-Notification. That is: If the value of the key is `true`, UDR notification with unmapped variables parsed inside **customAttributes** are forwarded to core service. If the value of the key is `false`, UDR notification with unmapped variables are forwarded to core service as it is received from UDR, without **customAttributes**. **Default value**: `true` **Note**: For PCRF Core to forward the custom attributes to Usage Monitoring service, you must set the value of ENABLE_CUSTOM_ATTRIBUTES_IN_UDR_NOTIFICATION to `false`.

Table 7-82 (Cont.) Parameters for Advanced Settings

Keys	Value
CREATE_PDSPROFILE_ON_REQUESTPC	Determines whether to create a PDS profile if it is not present. If the value is `true`, create request is sent if PDS Profile is empty. If the value is `false`, delete request is sent if PDS Profile is empty. **Default value**: `true`
PDS_ERROR_HANDLER_ENABLED	This flag enables error handling library integration with PDS service. **Default Value**: false
UPDATE_SSV_DATASOURCE_INFO_ON_CHANGE	This flag is used to avoid additional database operations (SEARCH and UPDATE) required to update the latest data source information received in the PATCH response. Possible values: • `true`: PolicyDS updates the latest data source information in the database received in the PATCH response for SSV entry. • `false`: PolicyDS skips updating the latest data source information in the database received in the PATCH response. **Default value**: `false`
USER.ocsSpendingLimit.resetContextOnGxCreate	When the value of this key is set to `true`, PCRF Core sets `resetContext` to `true` in PDS request reqParam for the first Gx session and PDS will revalidate the Sy session. **Default value**: `false`
UDR_FETCH_RELATED_RESOURCE	This flag fetches related UDR resource and forwards to UDR connector. **Default Value**: false
UPDATE_SSV_DATASOURCE_INFO_ON_CHANGE	When the value of this key is set to `true`, this flag updates SSV's subscription information following each PATCH call. The PATCH call returns updated subscription information. **Default Value**: false
NOTIFY_PCF_SM_ON_SUBSCRIPTION_FAILURE	Indicates whether to notify the SM service if there is a subscription failure. **Default Value**: true
NOTIFY_PCF_UE_ON_SUBSCRIPTION_FAILURE	Indicates whether to notify the UE Policy service if there is a subscription failure. **Default Value**: true
NOTIFY_PCF_AM_ON_SUBSCRIPTION_FAILURE	Indicates whether to notify the AM service if there is a subscription failure. **Default Value**: true
SUBSCRIPTION_FAILURE_NOTIFICATION_DELAY	Indicates the delay for the initial subscription failure notification (in milli seconds) **Default Value**: 3000

Table 7-82 (Cont.) Parameters for Advanced Settings

Keys	Value
SUBSCRIPTION_FAILURE_NOTIFICATION_RETRY_COUNT	Indiates the retry count for subscription failure notification (if value is 3 then maximum 3 calls will be done including first web call). **Default Value**: 3
SUBSCRIPTION_FAILURE_NOTIFICATION_RETRY_DELAY	Indicates the delay between retry of subscription failure notification. **Default Value**: 3000
PDS_NOTIFY_USER_DATA_REQUEST_PRIORITY	Set request priority for Notify user data requests. **Allowed Values**: 0-100 **Default Value**: 18
PDS_GET_DEFAULT_WORKFLOW_REQUEST_PRIORITY	Set request priority to default PDS workflow request. **Allowed Values**: 0-100 **Default Value**: 30
PDS_GET_USER_DATA_REQUEST_PRIORITY	Set request priority to fetch User data request. **Allowed Values**: 0-100 **Default Value**: 18
PDS_UPDATE_USER_DATA_REQUEST_PRIORITY	Set request priority to update User data for request. **Allowed Values**: 0-100 **Default Value**: 18
PDS_DELETE_USER_DATA_REQUEST_PRIORITY	Set request priority to delete User data request. **Allowed Values**: 0-100 **Default Value**: 16
PDS_AUDIT_NOTIFY_REQUEST_PRIORITY	Set request priority to Audit notify request. **Allowed Values**: 0-100 **Default Value**: 27
CONGESTION_RESPONSE_CODE	Configure the response code of the rejected requests by PDS due to congestion state. By default, PDS sends 503 as response code. If configured then it shall be 5xx only. **Note**: For Policy application with Aspen Serivce Mesh (ASM) setups, add the `CONGESTION_RESPONSE_CODE` set to 500 using PDS Advanced Settings. **Allowed Values**: 5xx **Default Value**: 503
ENABLE_STACK_TRACE_FOR_DB_OPS	When this flag is enabled, it prints the complete exception stack trace. When this flag is disabled, it prints the exception message rather than the complete stack trace. **Default Value**: false

Table 7-82 (Cont.) Parameters for Advanced Settings

Keys	Value
CONCURRENCY.LOCK_RETRY_MODE	Indicates whether the lock retry must be triggered under PDS. Possible valures are: • **CLIENT_ONLY**: lock retry is triggered only in client services. • **SERVER_ONLY**: lock retry is triggered only in bulwark. • **CLIENT_AND_SERVER**: lock retry is triggered by both server and client.(server will retry to acquire the lock till the lockWaitTimeout is expired and client will retry after the configured request backOff timer is expired) • Consumer services's advance settings should be updated with the non-zero lockWaitTimeOut.

8. Click **Save**.
   The page saves the PDS settings.

7.3.11.2 PDS Workflow

Using the PDS Workflow page, you can alter the behavior of PDS to suit the network requirements where you are deploying Policy. This page displays the call flow, depending on the value you select for Workflow type and the mode of deployment.

On upgrading to Policy, the system comes up with the default workflow.

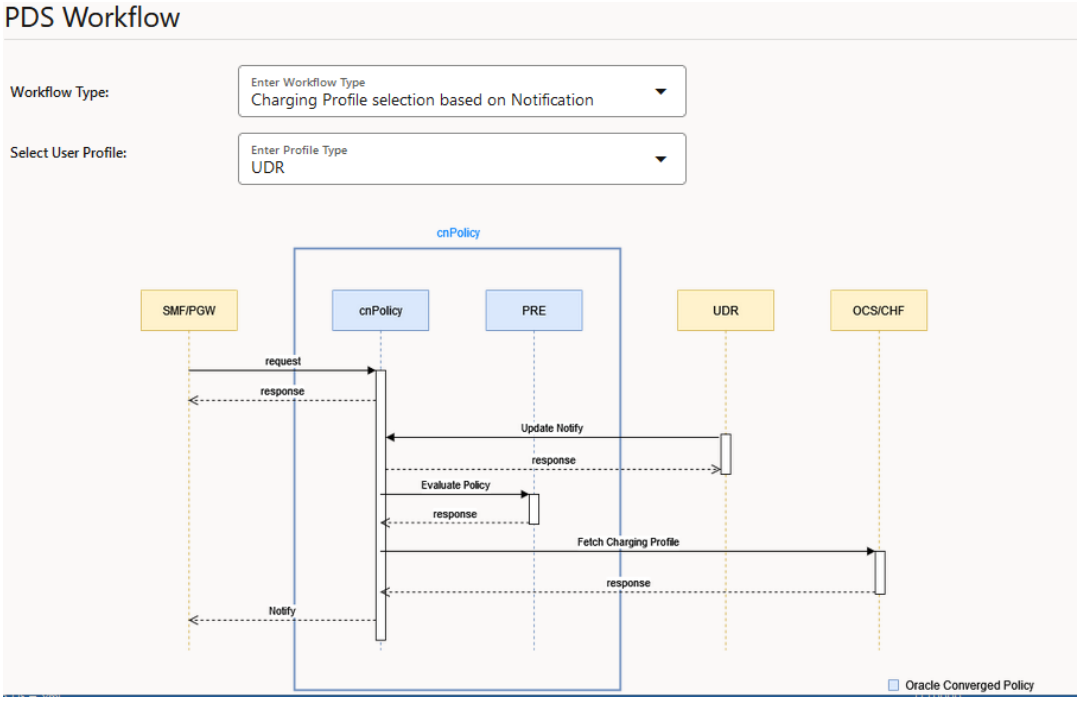
To change the PDS workflow, perform the following steps:

1. On the PDS Workflow page, click Edit.
2. Select any of the following values from the drop-down list:

Table 7-83 Workflow Type

Value	Description
Default	On selecting this option, the default PDS behavior applies.
Charging Profile selection based on User Profile	On selecting this option, you can configure whether to select user profile from LDAP or UDR. **Note**: For Sy selection per subscriber on the basis of LDAP attribute, select the value LDAP for user profile.
Charging Profile selection based on Notification	On selecting this option, you can configure to select user profile based on Notification from UDR.
Datasource selection based on PRE	On selecting this option, you can configure to select datasource based on PRE.
Custom	For a custom workflow, contact Oracle Policy engineering team and paste the workflow in JSON format provided by the team.

Figure 7-18 PDS Workflow



3. Click **Save** to save your updates.

7.3.12 Binding Service

This procedure provides information about configuring the Binding Service.

The **Binding Service Configuration** page displays the configurations for the Binding Service. You can edit the existing configurations using this page.

To configure the Binding service:

1. From the navigation menu under **Policy**, navigate to **Service Configurations**, and select **Binding Service**.
   This opens the **Binding Service Configuration** page. The page displays the existing configurations.
2. Click Edit .
   This opens the **Edit Binding Service Configuration** page.
3. Check the default configuration for all the fields in all groups and edit as necessary.
   The following table describes the input fields displayed under each group:

Table 7-84 Edit Binding Service Configuration

Field Name	Description
BSF Enabled	Determines if BSF is enabled or disabled. When the the value of the **Binding Operation** field is set to false, BSF cannot be enabled.
Binding Use Local Configured Bsf Always	Specifies whether to use local configured BSF without Always discovering. **Default Value**: FALSE

Table 7-84 (Cont.) Edit Binding Service Configuration

Field Name	Description
Binding Use Local Configured Bsf When Not Discovered	Whether to use local configured (if having) BSF when not discovered or discover failed. Local configuration can be done using custom yaml. **Default Value**: FALSE
Use HTTP2	Determines if using http/2 to communicate with BSF. Otherwise use http/1.1. **Default Value** : TRUE
Abort or Terminate Session on BSF Error	Determines if PCF (SM service) should abort terminate session when binding error is received. **Note**: This works for Create, Modify, and Pending Transactions requests only.
Session Limit By User	This parameter can be used to enable or disable user level limiting for bindings at Binding service. Binding service only marks the sessions as stale, and then notifies either SM or PCRF Core to perform the clean-up. The default value for this parameter is true.
Max Session Limit By User	This parameter can be used to specify the maximum session limit for a particular user. The default value for this parameter is 10.
Max Session Limit By APN	This parameter can be used to enable or disable limiting of number of bindings at DNN + SNSSAI + IP domain level at Binding service. The default value for this parameter is true.
Max Session Cleanup Mode	Binding service uses this flag to send request to SM service for either local or remote session clean up. The allowed values for this flag are: • `Local` • `Remote` On choosing the flag to `Local`, PCF deletes the session locally and does not send notify terminate request toward SMF. **Default Value**: Local
Max Session Force Delete On Expiry Of Wait Timer For Remote Cleanup	On `Max Session Cleanup Mode` set to `Remote` this toggle flag appears. It makes `forceDeleteOnExpiryOfWaitTimer` flag to true or false which is used in SM service request for remote cleanup. **Default Value**: false
Max Session Force Delete On Error For Remote Cleanup	On `Max Session Cleanup Mode` set to `Remote` this toggle flag appears. It makes `forceDeleteOnError` flag to true or false which is used in SM service request for remote cleanup. On SM service notify terminate request failure and based on `forceDeleteOnError` flag, the session with 4xx and 5xx response status codes are deleted except for the 404 status code. **Default Value**: false

Table 7-84 (Cont.) Edit Binding Service Configuration

Field Name	Description
Apn Filter By Snssai	This parameter can be used to specify the maximum session limit for DNN + SNSSAI + IP domain level. The default value for this parameter is 2.
BSF Retry Profile For Initial Messages	Retry Profile to be used when PCF fails to send a create message to a producer node.
BSF Retry Profile For Subsequent Messages	Retry Profile to be used when PCF fails to send an in-session message to a producer node. **Note:** • If both “Retry Profile for Initial Messages” and “Retry Profile for Subsequent Messages” are not configured (default case) no retry is attempted. • If both “Retry Profile for Initial Messages” and “Retry Profile for Subsequent Messages” are configured, retries is attempted accordingly. • If “Retry Profile for Initial Messages” is configured but “Retry Profile for Subsequent Messages” is not configured then the profile for initial messages is used for subsequent messages. (to be backward compatible with release 1.8.x) • If “Retry Profile for Initial Messages” is not configured but “Retry Profile for Subsequent Messages” is configured then retry is not attempted for initial messages but is attempted for subsequent messages.
BSF Communication Profile	Specifies the NF communication profile used by BSF. For more information about configuring NF communication profiles, see NF Communication Profiles.
Enable Vendor Specific Attribute in Register Request	Determines if PCF sends Vendor-Specific-Attribute in the Binding Create or Register request. By default, the value is set to false.
Vendor ID	**Note**: This field becomes active only when **Enable Vendor Specific Attribute in Register Request** switch is enabled. Specifies the VendorI ID that is mapped to the `vendorId` in the Vendor-Specific-Attribute.
Data Compression Scheme	Specifies the data compression scheme used by Binding Service. Following are the allowed values for this field: • Disabled • Zlib_Compressed By default, this value remains disabled.

4. Expand the **Audit** group.
   This group allows you to edit Audit configurations.
5. Enter the values for the input fields, available under the group.

Table 7-85 Edit Audit Configurations

Field Name	Description
Enabled	Determines whether to send registration request to Audit service or not. **Default Value**: True
Notification Rate (per second)	Defines the number of stale records which Audit service notifies to Binding service in one second. **Default and Recommended Value**: 50 **Note**: To configure higher number than the recommended value, contact My Oracle Support (https://support.oracle.com)
Binding Age (in minutes)	Defines the age of the binding entry after which Binding service queries the SM service or the PCRF Core service to check the staleness. **Default Value**: 1500
Maximum Binding Age (in minutes)	Defines the age of the binding entry after which records are deleted from the binding without querying the SM or PCRF Core services. **Default Value**: 2940
Minimum Audit Attempts	Specifies the minimum number of consecutive failed audit attempts until maxTTL / forceTTL is reached. If maxTTL is reached and audit_attempts + 1 >= Minimum Audit Attempts for maxTTL, Audit service sends notification to Binding Service with maxTTL flag set to *true*. Binding Service deletes the corresponding contextbinding information from its database. **Range**: 0-255 **Default Value**: 0 **Note**: If maxTTL is not reached and if audit attempts are reached, the number of audit attempts are incremented until maxTTL is reached.
Minimum Audit Passes Interval (in minutes)	Defines the time when next audit for the Binding service table is done after delta time if auditing this table has been finished before this specified time. **Default Value**: 15

6. Click **Save**.
   The page saves the Binding Service configurations.
7. Perform the following steps to configure **Advanced Settings** that sets the key for configuring congestion response code and message priority to various rest api end points:
   a. Click the ⊕ Add
      The page opens the **Add Advanced Settings** dialog box.
   b. In the dialog box, enter the following **key** and respective **value**:

Table 7-86 Parameters for Advanced Settings

Keys	Value
CONGESTION_RESPONSE_CODE	Configure the response code of the any rejected requests by Usage Monitoring due to congestion state. By default, Binding service sends 503 as response code. If configured then it shall be 5xx only. **Allowed Values**: 5xx **Default Value**: 503
AUDIT_MESSAGE_PRIORITY	Set request priority for Audit notification request. **Allowed Values**: 0-100 **Default Value**: 30
BSF_AUDIT_MESSAGE_PRIORITY	Set request priority for BSF Audit request. **Allowed Values**: 0-100 **Default Value**: 27
DEPENDENT_CONTEXT_BINDING_REGISTER_MESSAGE_PRIORITY	Set request priority for registring binding contextbinidng request. **Allowed Values**: 0-100 **Default Value**: 24
DEPENDENT_CONTEXT_BINDING_DEREGISTER_MESSAGE_PRIORITY	Set request priority for deregistering binding contextbinding request. **Allowed Values**: 0-100 **Default Value**: 16
DEPENDENT_CONTEXT_BINDING_FIND_CONTEXT_OWNER_MESSAGE_PRIORITY	Set request priority for finding dependent contextbinidng owner request. **Allowed Values**: 0-100 **Default Value**: 16
SESSION_BINDING_REGISTER_MESSAGE_PRIORITY	Set request priority for session binding register request. **Allowed Values**: 0-100 **Default Value**: 24
SESSION_BINDING_UPDATE_MESSAGE_PRIORITY	Set request priority for session binding update request . **Allowed Values**: 0-100 **Default Value**: 20
SESSION_BINDING_DEREGISTER_MESSAGE_PRIORITY	Set request priority for deregistering session binding request. **Allowed Values**: 0-100 **Default Value**: 16
SESSION_BINDING_SEARCH_MESSAGE_PRIORITY	Set request priority for searcing session binding request. **Allowed Values**: 0-100 **Default Value**: 30
SESSION_BINDING_FIND_CONTEXT_OWNER_MESSAGE_PRIORITY	Set request priority for finding binding context owner request. **Allowed Values**: 0-100 **Default Value**: 16
SESSION_BINDING_CLEANUP_MESSAGE_PRIORITY	Set request priority for session binding cleanup request. **Allowed Values**: 0-100 **Default Value**: 30

Table 7-86 (Cont.) Parameters for Advanced Settings

Keys	Value
BSF_SESSION_UPDATE_MESSAGE_PRIORITY	Set request priority for binding session update. **Allowed Values**: 0-100 **Default Value**: 20
STALE_SESSION_TRACKER_REFRESH_MESSAGE_PRIORITY	Set request priority for tracking stale session request. **Allowed Values**: 0-100 **Default Value**: 30

7.3.13 Policy Engine

This section provides information about viewing the Policy service details.

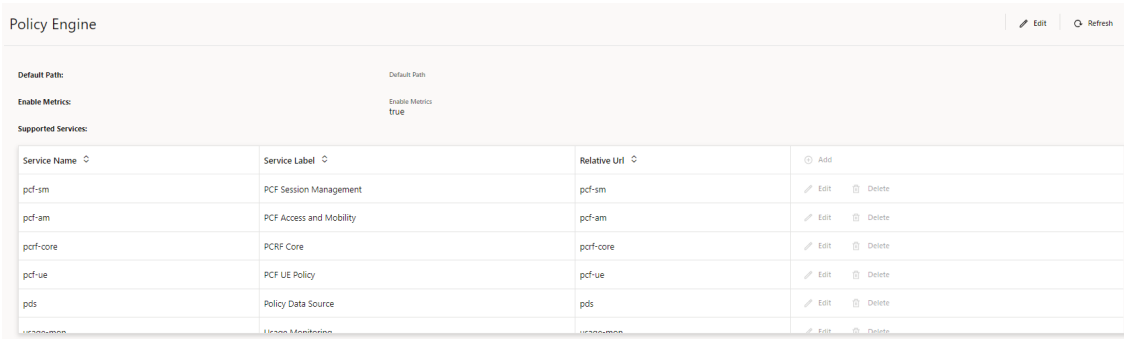
The **Policy Engine** page displays the list of the services available on the Policy deployment.

To access **Policy Engine** page from CNC Console home page, expand **Policy**, navigate to **Service Configurations** and select **Policy Engine**.

> **Note**
>
> This page must only be used for adding the Policy Data Source (PDS) service. Rest of the services must not be added or edited using this page.

The following screen capture shows an illustrations of the **Policy Engine** page:

Figure 7-19 Policy Engine



Configuring PDS Service

To add or configure PDS service:

1. On the **Policy Engine** page, click Edit.
   This opens the **Edit PDS Service** page displaying the supported services in a tabular format, under the **Supported Services** group.
2. Make sure that the Enable Metrics switch is enabled. This field specifies whether to enable the metrics for this functionality. By default, this switch remains disabled.

3. Click Add available in the table.
   The page opens the **Add Supported Services** dialog box.
4. Enter the following information:

Table 7-87 Add Supported Services Configuration

Field Name	Description
Service Name	The name of the service. For PDS, enter the value: `pds`
Service Label	The service label. Enter `Policy Data Source`
Relative Url	Enter `pds`

5. Click **Save** on the dialog box.
   The **pds** service gets listed on the **Edit PDS Service** page.

> **Note**
>
> Use or available under the **Actions** column to update or delete the pds service details.

6. On the **Edit PDS Service** page, click **Save**.
   The page saves PDS as the supported service.

7.3.14 Configuring NWDAF Agent

This section provides information on how to configure NWDAF Agent.

7.3.14.1 Settings

This procedure provides information about managing NWDAF Agent Settings

The Settings page allows you to edit Advanced Settings for NWDAF Agent service.

To configure, perform the following steps:

1. From the navigation menu, under **Policy**, click **Service Configurations**, click **NWDAF Agent**, and then select **Settings**.
   This opens the Settings page.
2. Click **Edit**.

   This opens the Edit NWDAF Agent Settings page.
3. Click **Add**.
4. On the **Add Advanced Settings** dialog box, enter values for the following input fields:
   - **Key** - Specify the name of the key.
   - **Value** - Specify the value for the key.
5. Click **Save** to save the advanced settings.
   To discard the changes, click **Cancel**
6. Click **Save** to save the NWDAF Agent Settings.
   To discard the changes, click **Cancel**

7.3.14.2 Slice Load Level

This procedure provides information on how to add and manage Slice Load Level.

The Slice Load Level page allows you to add and edit Slice Load Levels.

To configure, perform the following steps:

1. From the navigation menu, under **Policy**, click **Service Configurations**, click **NWDAF Agent**, click **Autonomous Subscription**, and then select **Slice Load Level**.
   This opens the Slice Load Level Configuration page.
2. Click **Add**.

   This opens the Config Slice Load Level page.
3. Enter values for the following input fields, as described in the following table:

Table 7-88 Config Slice Load Level

Field Name	Description
SNSSAI	Identification of network slice to which the subscription applies.
Threshold	Indication of slice load level in percentage.
ImmRep	Indication of immediate reporting.

4. Click **Save**. The Slice Load Level is displayed on the Slice Load Level Configuration page.
   To discard the changes, click **Cancel**

Importing Slice Load Level

To import Slice Load Level :

1. Click 

   .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.3.15 NRF Agent

Most of the PCF services discovery requests(On Demand Discovery) towards NRF do not use UE identifiers for NF discovery. Hence the requests do not carry correlation-info header. But On Demand Discovery of UDRs requests have the UE identifiers like SUPI or GPSI. These UDR discovery request with UE identifier toward NRF is updated to support correlation-info header.

Policy supports correlation-info header toward NRF by using the **NRF Agent** page in the CNC Console. The NF communication profiles configured for NRF agent, generates or forwards the correlation-header that is used to for NRF discovery requests. To configure NRF Agent Communication profile, perform the following steps:

1. From the navigation menu under **Policy,** navigate to **Service Configurations**, and then select **NRF Agent**. This opens the **NRF Agent** page.
2. Click Edit. This opens **Edit NRF Agent** page.
3. On the **Edit NRF Agent** page, enter the following information:

Table 7-89 Edit NRF Agent

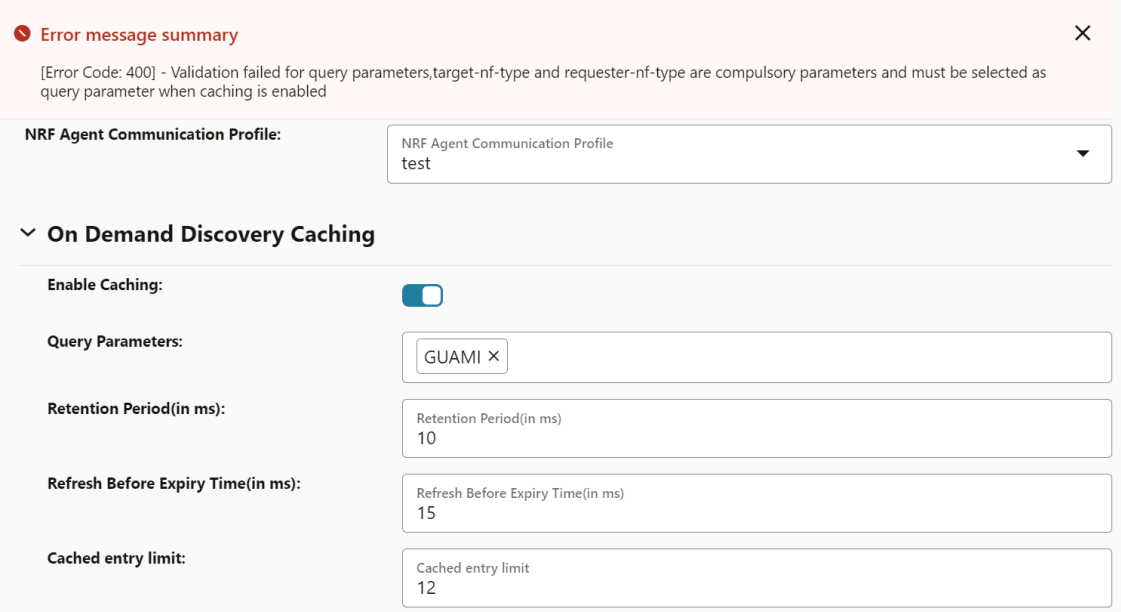
Field Name	Description
NRF Agent Communication Profile	Specifies the NRF Communication Profile to handle correlation info header toward NRF.
On Demand Discovery Caching	**Provide details about On Demand Discovery Caching**
Enable Caching	Indicates NRF Client to cache the profile data and use it to process the future discovery on demand requests. **Default value**: false
Query Parameters	Accepts the list of parameters that NRF Client uses to fetch the profile data from cache. **Note**: The query parameters in the request from SM service, AM service, UE Policy service must match and must be in the same order as the data received from NRF. Otherwise, NRF Client can not process the request. Supported values: • data-set • target-nf-type • requester-nf-type • target-nf-set-id • service-names • amf-region-id • amf-set-id • GUAMI • target-nf-instance-id • group-id-list
Retention Period (in ms)	Indicates the time a record is allowed to stay in database after expiry and before any clean process is executed. **Default value**: 864000000
Refresh Before Expiry Time (in ms)	Indicates the time in milliseconds before expiration when a rediscovery must be performed for the requested discovery parameters. **Default value**: 4500
Cached entry limit	Parameter to set the upper bound of cache registries allowed to be retained. **Default value**: 2000
Auditor Scheduler Interval (in ms)	Parameter to set time in milliseconds to activate the Discovery Cache auditor service to clean stale records. **Default value**: 1800000
Auditor Purge Percentage	Parameter to set the percentage of the cached discovery requests that will be pruned once the threshold has been passed. **Default value:** 20

4. Click **Save** to save the changes

When selecting the query parameters to perform on demand discovery by enabling NRF caching, ensure that `target-nf-type` and `requester-nf-type` parameters are selected along

with other query parameters. If they are not selected then the configurations are not saved and error message is displayed on the **NRF Agent** page.

Figure 7-20 NRF Agent Error Message



7.3.16 IGW

1. From the navigation menu under **Policy,** navigate to **Service Configurations**, and then select **IGW**. This opens the **Error Code Series List** page.
2. Click Edit on the Error Code Series List.

   This opens **Edit Error Code Series List** page.
3. Click **Add** to add the error code series.

   The **Add Error Code Series** page is displayed.
4. Configure the fields in the **Add Error Code Series** page as mentioned in Table 7-90.
5. Click **Add** in the **Error Code Series** section.

   The **Add Error Code Series** page is displayed.
6. Configure the fields in the **Add Error Code Series** page as mentioned in Table 7-91.
7. Click **Save** on the **Add Error Code Series** page to save the details. Click **Cancel** to discard your progress and go back to **Create Error Code Series** page.
8. Click **Save** on the **Create Error Code Series**page to save the details. Click **Cancel** to discard your progress and go back to **Error Code Series** page.

> **Note**
>
> Use the **Edit** icon or **Delete** available in the next column of the specific entry to update or delete the error code series information.

Table 7-90 Error Code Series

Field Name	Description
Profile	Specifies the name for error code series list profile.
Error Code Series	Lists the error codes for a specific service.

Table 7-91 Error Code Series Configuration

Field Name	Description
Error Set	Possible values for "errorSet" attribute: 5xx, 4xx, 3xx, 2xx, 1xx
Error Codes	Possible values include all error codes in the respective HttpSeries value assigned for "errorSet". **Note**: Use single value of "-1" if all error codes in that HttpSeries are to be considered.

7.3.17 Bulwark

This section describes how to configure Bulwark service.

7.3.17.1 Bulwark Settings

This procedure provides information about managing Bulwark service settings.

To configure, perform the following steps:

1. From the navigation menu, under **Policy**, click **Service Configurations**, click **Bulwark**, and then select **Settings**.
   This opens the Bulwark Settings page.
2. Click **Edit**.

   This opens the Edit Bulwark Settings page.
3. Under **Server Retry On Already Locked Keys** section, configure the following fields:

   Table 7-92 Server Retry On Already Locked Keys

Field	Description
Enable	Indicates whether to enable or disable server retry on already locked keys. To enable the feature, set the value of this field to `true`. **Default value**: `false`
Maximum Pending Lock Requests allowed per Key	Indicates the maximum number of pending lock requests allowed per key. **Default value**: 5
4. Click **Save** to save the settings.
   To discard the changes, click **Cancel**

7.4 Policy Data Configurations

This section describes how to create the manageable objects (MOs) in Policy, using the **Policy Data Configurations** pages.

7.4.1 Common

This section includes the common services configurations for Policy services.

To access Common Data functionality from the CNC Console home page, expand **Policy**, navigate to **Policy Data Configurations** and select **Common**.

The common configurations include:

- Policy Table
- Dropdown Blocks
- PCF Presence Reporting Area
- Policy Counter ID
- Match List
- Schemas

7.4.1.1 Policy Table

Policy Tables are the independent objects that can be used to capture differences in policy structures.

Most of the policies are very similar with small differences between them. A policy table abstracts the differences between related policies. Instead of creating many similar policies, using a policy table makes the tasks of adding new policies, modifying existing sets of policies, and checking consistency among related policies simpler and less prone to error.

A Policy table resembles the database tables and consists of the following elements:

- Table name
- Table description
- Column definitions
  Every column has a definition that contains a name, data type, and indication if the column is a key column. Every entry in the column is of the same data type as the column. Every policy table must have at least one key column.
- Data
  The contents of the table cells. Blank cells are not allowed in a policy table.

Each row in a policy table can be thought of as a scenario. Substitutions in policy condition and action parameters can include the values in a specified policy table.

> **Note**
>
> CNC Policy supports Policy Table for the following services:
>
> - SM
> - AM
> - UE Policy
> - PCRF Core
> - Usage Monitoring

You can manage multiple policies with small differences by abstracting the differences into tables. This makes the process of modifying the policies, or creating new or similar policies a matter of modifying the policy table, which is simpler and less prone to error.

Managing Policy Tables

This section describes how to create, modify, delete, and view policy tables.

To access Policy Table configurations from the CNC Console home page, expand **Policy**, navigate to **Policy Data Configurations** and select **Policy Table**.

The **Policy Table** page allows you to perform the following tasks:

- Configure Policy Table
- Associating a Policy Table with a Policy

7.4.1.1.1 Configure Policy Table

This procedure provides information on how to create and manage Policy Tables.

The **Policy Tables** page allows you to create new and manage existing policy tables. The page displays the list of available policy tables and provides the options to create, configure, import, export, clone, modify, or delete the tables.

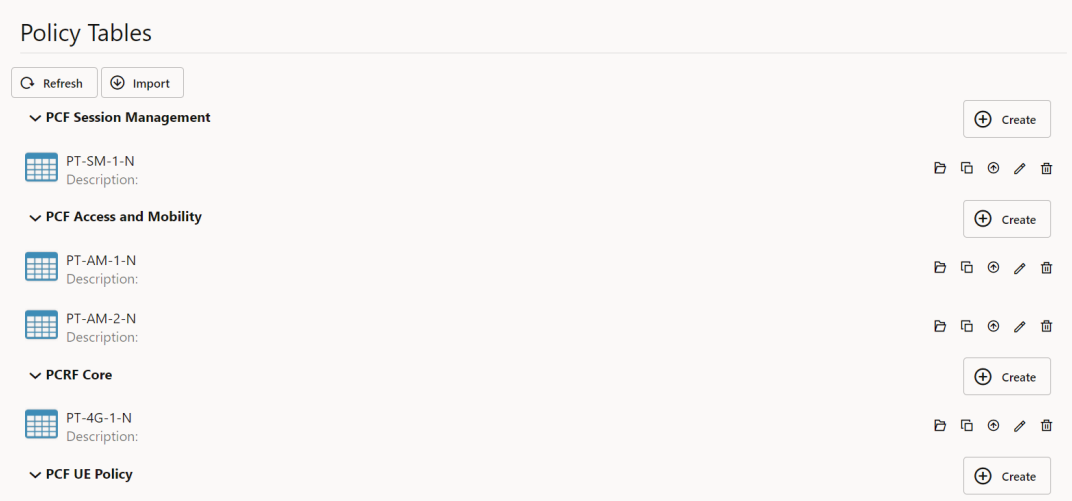
> **Important**
>
> When you define a policy table, it must contain at least one key column and one row, and you must populate every cell in the table.

To create and configure a policy table, perform the following steps:

1. From the **Policy Management** section of the navigation pane, select **Policy Table**.

   This opens the **Policy Tables** page, displaying the existing policy tables for the respective policy service.
   The following screen capture shows an illustration of the **Policy Tables** page:

Figure 7-21 Policy Tables



2. Click Create for the service group, for which the policy table is to be created.

   The page opens the **Create Policy Table** dialog box.

3. In the **Create Policy Table** dialog box, enter the following information:

Field Name	Description
Name	The unique name you assign to the policy table. This is a mandatory field. The name can only contain the characters A–Z, a–z, 0–9, period (.), hyphen (-), and underscore (_). The maximum length is 32 characters.
Description	Free-form text that identifies the policy table. The maximum length is 255 characters.

> **Note**
>
> Name can not be the substring of any other configured Policy Table Name.

4. Click **Save**.

   This creates the Policy Table. The table gets listed under the respective service name group.

> **Note**
>
> - You can create maximum 80 tables per service type.
> - The **Policy Table** page allows you to clone policy tables. Use the icon available with the table listing, to clone a policy table. The cloned table contains the data similar to the table from which it is cloned.
> - Use or available with the table listing, to update or delete the policy table.

5. To configure the policy table, click corresponding to the policy table name.

   This opens a new page for the selected policy table. You can configure the table rows and columns using this page.

6. Click Create Column .

   The page opens the **Create Policy Table Column** dialog box.

   > **Note**
   >
   > You must define at least one key column. You can define maximum 15 key columns in a policy table.

   Enter the information as appropriate:

Field Name	Description
Name	The name you assign to the column. This is a mandatory field. The name can only contain the characters A–Z, a–z, 0–9, space ( ), and underscore (_). The maximum length is 32 characters. **Note:** Column Name must be unique.
Data Type	The data type of cells in the column. This is a mandatory field. You can select the required data type from the drop-down list.
Key	Enable this switch if this is a key column. **Note**: By default, the first column is always the **Key** column.

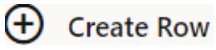
   > **Note**
   >
   > Name can not be the substring of any other configured Column Name

7. Click **Save**.

> **Note**
>
> You can create maximum 30 columns in a policy table.

This creates the column for the policy table. You can use the  or  to update or delete the columns.

> **Note**
>
> Consider the following points while adding a new column in a policy table:
>
> - The Add, Modify, or Delete operations are allowed for both key and non-key columns, irrespective of the fact whether the policy table contains row(s) or not.
> - A key column can be created for a policy table containing row(s).
> - The row values for the corresponding column and its data type have Empty/ Null Values, you have to manually update the rows, except for boolean data type, which is set to false.
> - While modifying a column, the **Data Type** field remains disabled, if the policy table contain row(s).
> - While editing a column, a warning message is displayed to revisit the policies after editing them. You must agree to it by selecting the checkbox, to enable the **Save** button.
> - A warning is issued to update the policies manually, if the column is deleted.

8. (Optional) You can create rows in the policy table as follows:
   a. Click  .

      The page opens the **Create Policy Table Row** dialog box.
   b. Enter the values corresponding to each column of the policy table. The data in each cell must match with the respective column datatype.
   c. Click **Save**. You can also enter a comma-separated list of values in the **C1** column. Use the data type as array while entering comma-separated values.

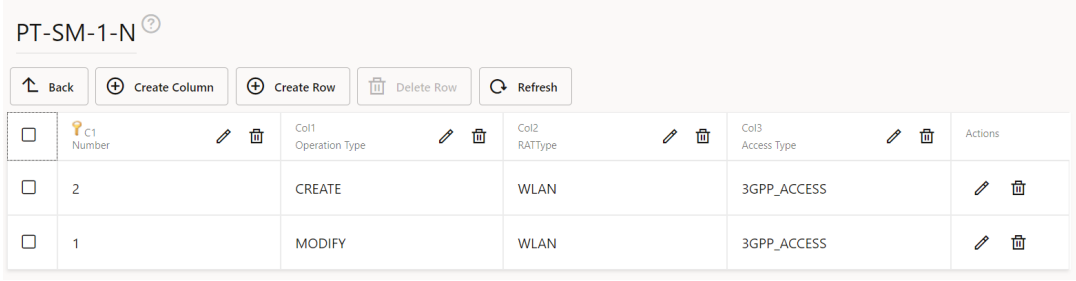
   This creates and display the row for the policy table. You can use  or  under the **Action** column to update or delete rows.

> **Note**
>
> - The policy table loads 100 rows in one fetch. As the user scrolls the page, next set of rows gets loaded and displayed.
> - You can create maximum 3000 rows in a policy table.
> - Make sure that the key column (**C1**) does not hold a combination of duplicate entries, that is, combination of two or more columns in a policy table can be used to uniquely identify each row.
> - One or more wild card values in the key column cell are compared to the values in the policy context data. If there is any match, the row gets matched. The asterisk (*) character represents any number of characters, and the question mark (?) character represents any single character.

The policy table is updated with columns and rows. You can now use the table in a policy.

The following screen capture illustrates an example of a Policy Table with four columns and two rows:

Figure 7-22 Policy Table with Rows and Columns



9. Click Back for returning to the **Policy Tables** page.

> **Note**
>
> he **Policy Table** page allows you to export policy tables. Use the icon available with the table listing, to export the policy table configuration in the JSON file format.

Importing Policy Table

To import a policy table:

1. Click Import .

   The page opens the **Upload Policy Table Data** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** option.
3. After the upload completes for the selected file, choose any of the following values from the **Conflict Resolution Strategy**:

Option	Description
Retain	On selecting the Retain option, no changes are made to the existing Policy Table. No changes are made to the UUIDs and existing policies remain unaffected with this option.
Replace	On selecting the Replace option, the following happens: • No change is made to UUIDs of existing columns. • UUIDs of existing rows is retained if the composite key of the existing row matches with the composite key of incoming row. • New key or non-key columns in the imported file are added to the existing Policy Table with new UUIDs. • Existing rows that are not present in imported file are deleted. **Note**: Column conflict is detected based on the column name and not data type. Using the replace option leads to updates in the Policy Table according to the data present in the import file. For instance, it may result in deletion or addition of key and non-key columns and their respective rows, and change in dataType. Thus, it is recommended to revisit the policies and update wherever required.
Merge	**Note**: Merge option is not allowed and results in an error in the following scenarios: • Change in Key Column: It includes scenarios when user adds a new column, removes an existing key column, or changes data type of existing key column. • Change in datatype of existing non-key column On selecting the Merge option, the following options are enabled: • **Retain**: In case of no conflict, new rows from the import file are created. However, in the case of conflict, no change is made to the cell values of the existing rows. In this option, new non-key columns are also added with their corresponding cell values.

Option	Description
	• **Replace**: In case of no conflict, new rows from the import file are created. However, in the case of conflict, existing cell values are replaced with the cell values of Policy Table in the import file. No change is made to the UUIDs of existing rows. In this option, new non-key columns are also added with their corresponding cell values. **Note**: Two rows are considered to be in conflict when they have same values for the Key columns. If the import file does not contain any existing non-key column, then that column is not removed from the existing Policy table during merge.

4. Click **Import**.

7.4.1.1.2 Associating a Policy Table with a Policy

This procedure provides information on how to associate a policy table with a new or existing policy.

To associate a policy table with a policy, the policy table must already be configured. For information about configuring Policy Tables, see Configure Policy Table.

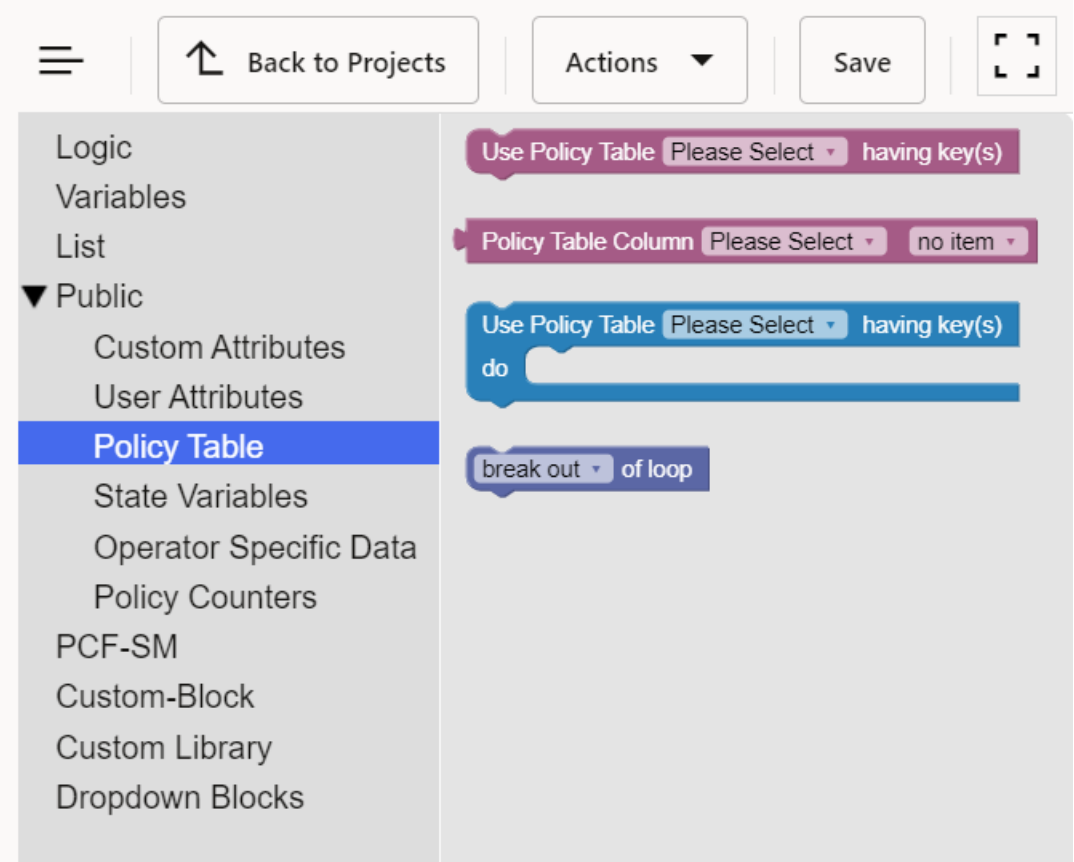
> **Note**
>
> When a user creates or updates a Policy Table through CNC Console or REST API, it may take up to 30 seconds for the changes to reflect in the policy rule associated with that Policy Table.

To associate a policy table with a policy rule:

1. From the navigation menu, under **Policy Management**, click **Policy Projects**.
   This opens the **Policy Projects** page displaying all the existing policies.
2. Click  corresponding to the policy to be associated.
   This opens a new page for the selected policy.
3. On the policy page, expand **Public**, and select **Policy Table**.
   This provides you the access to the Policy Table blocks.

   The following screen capture shows an example of Policy Table blocks for a **PCF Session Management** policy:

Figure 7-23 Example of Policy Table Blocks for SM Policy

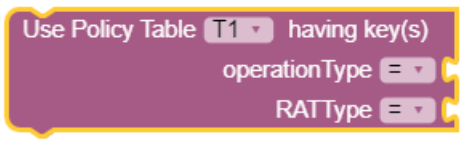


4. In the first block, select the policy table from the **Policy Table** drop-down list. The corresponding key columns are displayed in **key(s)**.

   The following screen capture shows an example in which Policy Table **T1** is selected and the **OperationType** and **RatType** are the corresponding key columns in the table **T1**:

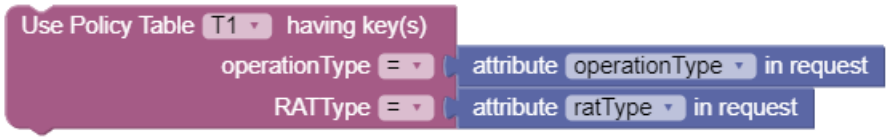
> **Note**
>
> This block is being deprecated. Use the third block on the **Policy Tables** page to achieve the functionality of the deprecated block. For more information, see Step 7.

Figure 7-24 Example of Policy Table First Block



5. Select the operator from the operator drop-down list and associate the value or policy condition with the key column. You can select the value or policy condition from **Public** and **PCF-SM** topics. .
   The following screen capture shows an example of associating policy conditions with the key columns, **OperationType** and **RatType**:

Figure 7-25 Example of Associating Policy Conditions with Key Columns



If all the values associated with the key columns match its column data from the policy table based on the used operator ("="), then it returns the complete row data.

> **Note**
>
> This block can return only single row.

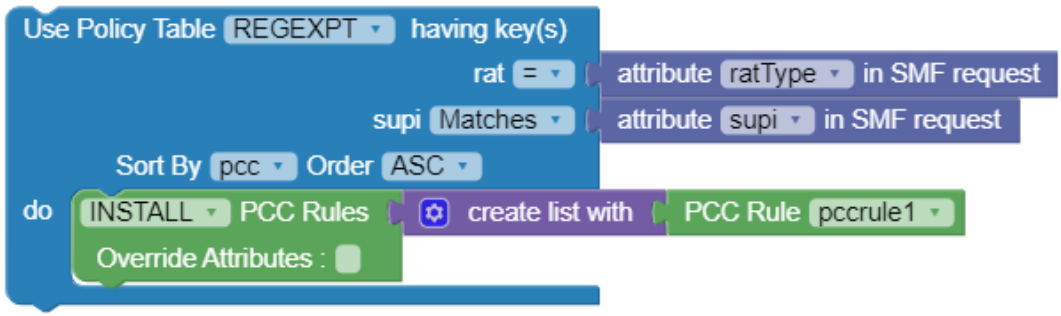
6. In the second block, select the policy table from the **Policy Table Column** drop-down and the corresponding non-key columns are displayed in **no item** drop-down list.
The following screen capture shows an example in which policy table **T1** is selected and the non-key column, **pccRule** is displayed in the drop-down list:

Figure 7-26 Example of Policy Table with non-key Column

This block returns the value of the non-key column selected by taking row data as input from the first block.

7. In the third block, select the policy table from the **Policy Table** drop-down list and the corresponding key and non-key columns are displayed in **key(s)**.
The following screen capture shows an example in which Policy Table **REGEXPT** has been selected and the **rat** and **supi** are the corresponding columns in the table **REGEXPT**.

Figure 7-27 Example of Policy Table with Corresponding Columns



Select the operator from the operator drop-down and associate the value or policy condition with the columns. You can select the value or policy condition from **Public** and

PCF-SM topics. The following screen capture shows an example of associating policy conditions with the columns, **RAT** and **SUPI**.

If all the values associated with the columns match its column data from policy table based on the operator used ("=") and Matches, then it returns the complete row data.

> **Note**
>
> - This block can return multiple rows.
> - For this block, "=", "!=", Matches, and Ignore opearotrs are supported.

8. Click **Save**.

The selected policy tables get associated with this policy rule.

7.4.1.2 Dropdown Blocks

This procedure provides information about how to create and manage dropdown blocks for policy projects.

The **Dropdown Blocks** page allows you to create new and manage existing blocks. The page displays the list of defined dropdown blocks with the options to import, export, or add blocks.

To configure the Dropdown blocks:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **Common**, and select **Dropdown Blocks**.
   This opens the **Dropdown Blocks** page. The page lists the existing dropdown blocks. You can add or import new dropdown block using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Dropdown Blocks** page.
3. On the **Create Dropdown Blocks** page, enter the following information:

Table 7-93 Create Dropdown Blocks

Field Name	Description
Attribute Name	Name of the attribute
Description	Description of the attribute
Type	Select from the following values: • static • dynamic

4. Expand the **Block Options** group.
   This group allows you to add multiple block options.
5. To add block options:

a. Click Add .
The page opens the **Add Block Options** dialog box.

b. On the dialog box, enter the following values:

Table 7-94 Add Block Options

Field Name	Description
Label Name	Name of the block
Value	Specify the value for the block option.

c. Click **Save**.
The block option gets listed under the **Block Option** group on the **Create DropDown Blocks** page.

6. On the **Create DropDown Blocks** page, click **Save**.
The page lists the dropdown block.

> **Note**
>
> Use or available under the **Actions** column to update or delete the dropdown block.

Importing Dropdown Blocks

To import Dropdown Blocks:

1. Click Import .
The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.1.3 PCF Presence Reporting Area

This procedure provides information about how to create and manage PCF Presence Reporting Area (PRA).

The **PCF Presence Reporting Area** page allows you to create new and manage existing PRAs. The page displays the list of defined PRAs and provides the options to import, export, or add PRAs.

To configure PRAs:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **Common**, and then select **PCF Presence Reporting Area**.
This opens the **PCF Presence Reporting Area** page. The page lists the existing PRAs listed in a tabular format. You can add or import new PRAs using this page.

> **Note**
>
> Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create PCF Presence Reporting Area** page.
3. On the **Create PCF Presence Reporting Area** page, enter the following information:

Table 7-95 Create PCF Presence Reporting Area

Field Name	Description
Pra Id	The unique identifying number of the PRA list. The ID must be a number between 0 and 16777125. This field is present if the Area of Interest subscribed or reported is a Presence Reporting Area.
Name	The unique name assigned to the PRA.

4. Expand the **Tracking Area List** group.
   This group allows you to add multiple Tracking Area List
5. To add the Tracking Area Lists:
   a. Click Add .
      The page opens the **Add Tracking Area List** dialog box.
   b. On the dialog box, enter the following values:

Table 7-96 Add Tracking Area List

Field Name	Description
PLMN Id	
MNC	Defines the Mobile Network Code. It can be two or three digit number.
MCC	Defines the Mobile Country Code. It should be a 3-digit number.
TAC	28-bit string identifying an E-UTRAN Cell Id as specified, in hexadecimal representation. Each character in the string takes a value of "0" to "9" or "A" to "F" and represents 4 bits. The most significant character representing the 4 most significant bits of the Cell Id appears first in the string, and the character representing the 4 least significant bit of the Cell Id appears last in the string. Pattern: '^[A-Fa-f0-9]{7}$' Example: An E-UTRAN Cell Id 0x5BD6007 be encoded as "5BD6007".

   c. Click **Save**.
      The TAC gets listed under the **Tracking Area List** group on the **Create PCF Presence Reporting Area** page.
6. On the **Create PCF Presence Reporting Area** page, expand the **ECGI List** group.
   This group allows you to add multiple ECGI Lists.
7. To add the ECGI Lists:
   a. Click Add .
      The page opens the **Add ECGI List** dialog box.

b. On the dialog box, enter the following values:

Table 7-97 ECGI List

Field Name	Description
MNC	Defines the Mobile Network Code of the PLMN. It can be two or three digit number.
MCC	Defines the Mobile Country Code of the PLMN. It should be a 3-digit number.
Eutra Cell Id	28-bit string identifying an E-UTRA Cell Id as specified in hexadecimal representation. Each character in the string takes a value of "0" to "9" or "A" to "F" and represents 4 bits. The most significant character representing the 4 most significant bits of the Cell Id appears first in the string, and the character representing the 4 least significant bit of the Cell Id appears last in the string. Pattern: '^[A-Fa-f0-9]{7}$' Example: An E-UTRA Cell Id 0x5BD6007 be encoded as "5BD6007".

c. Click **Save**.
The ECGI List gets listed under the **ECGI List** group on the **Create PCF Presence Reporting Area** page.

8. On the **Create PCF Presence Reporting Area** page, expand the **NCGI List** group.
This group allows you to add multiple NCGI Lists.

9. To add the NCGI Lists:

a. Click Add .
The page opens the **Add NCGI List** dialog box.

b. On the dialog box, enter the following values:

Table 7-98 NCGI List

Field Name	Description
MNC	Defines the Mobile Network Code of the PLMN. It can be two or three digit number.
MCC	Defines the Mobile Country Code of the PLMN. It should be a 3-digit number.
NR Cell Id	36-bit string identifying an NR Cell Id as specified in hexadecimal representation. Each character in the string takes a value of "0" to "9" or "A" to "F" and represents 4 bits. The most significant character representing the 4 most significant bits of the Cell Id appears first in the string, and the character representing the 4 least significant bit of the Cell Id appears last in the string. Pattern: '^[A-Fa-f0-9]{9}$' Example: An NR Cell Id 0x225BD6007 is encoded as "225BD6007".

c. Click **Save**.

The NCGI List gets listed under the **NCGI List** group on the **Create PCF Presence Reporting Area** page.

10. On the **Create PCF Presence Reporting Area** page, expand the **Global RAN NodeId List** group.
    This group allows you to add multiple RAN Node ID Lists.
11. To add the Global RAN NodeId Lists:
    a. Click Add .
    The page opens the **Add Global RAN NodeId List** dialog box.
    b. On the dialog box, enter the following values:

Field Name	Description
PLMN Id	
MNC	Defines the Mobile Network Code of the PLMN. It can be a 2- or 3-digit number.
MCC	Defines the Mobile Country Code of the PLMN. It should be a 3-digit number.
N3 Iwf Id	This field is included if the RAN node belongs to non 3GPP access (i.e a N3IWF). It contains the FQDN of the N3IWF.
gNb Id	
Bit Length	Unsigned integer representing the bit length of the gNB ID within the range 22 to 32.
gNb Value	Specifies the gNb identifier. The value of the gNB ID is encoded in hexadecimal representation. Each character in the string takes a value of "0" to "9" or "A" to "F" and represents 4 bits. The most significant character representing the 4 most significant bits of the gNB ID appears first in the string, and the character representing the 4 least significant bit of the gNB ID appears last in the string. The string is formatted with following pattern: '^[A-Fa-f0-9]{6,8}$' Example: "382A3F47" indicates a gNB ID with value 0x382A3F47
Nge Nb Id	This field is included if the RAN Node Id represents a NG-eNB. It contains the identifier of an NG-eNB.

    c. Click **Save**.
    The Global RAN NodeId List gets listed under the **Global RAN NodeId List** group on the **Create PCF Presence Reporting Area** page.
12. On the **Create PCF Presence Reporting Area** page, click **Save**.
    The PRA gets listed on the **PCF Presence Reporting Area** page.

> **Note**
>
> Use or available under the **Actions** column to update or delete the PRA.

Importing PCF Presence Reporting Area

To import PRA:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.1.4 Policy Counter Id

This procedure provides information about how to create and manage Policy Counter Ids.

The **Policy Counter Id** page allows you to create new and manage existing IDs. The page displays the list of defined Policy Counter Ids and provides the options to import, export, or add new IDs.

To configure the Policy Counter Id:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **Common**, and then select **Policy Counter Id**.
   This opens the **Policy Counter Id** page. The page lists the existing policy counter IDs listed in a tabular format. You can add or import new IDs using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Policy Counter Id** page.
3. On the **Create Policy Counter Id** page, enter values for the following input fields:
   The following table describes the fields:

Table 7-99 Create Policy Counter Id

Field Name	Description
Policy Counter Id	Policy Counter Id's Name.
Name	The unique name of the counter ID.
Description	Policy Counter Id's description.
Default Status	

4. Click **Save**.
   The policy counter ID gets listed on the **Policy Counter Id** page.

   > **Note**
   >
   > Use or available under the **Actions** column to update or delete the counter ID.

Importing Policy Counter Id Data

To import Policy Counter Ids:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.1.5 Match Lists

In a wireless network, a match list is a set of values in various categories, including Access Point Names (APNs), subscriber IMSIs, Location Area Codes (LACs), Service Area Codes (SACs), Internet addresses, and user equipment identities. Match lists provide allowlist (listing items to be included) and blocklist (listing items to be excluded) functions in policy rules. Match lists support wildcards. Using wild cards, a range of values can be specified compactly.

By using a match list, you can apply a policy to all subscribers in a set of LACs, block access to a list of Internet addresses known to be high risk, and so on.

This procedure provides information about how to create and manage Match Lists.

The **Match List** page allows you to create new and manage existing lists. The page displays the list of defined match lists and provides the options to import, export, or add lists.

To confgure match lists:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **Common**, and then select **Match List**.
   This opens the **Match List** page. The page lists the existing match lists. You can add or import new lists using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Match List** page.
3. Enter values for the input fields available on the page:
   The following table describes the fields:

Table 7-100 Create Match List

Field Name	Description
ID	The ID assigned to the match list.
Name	The name assigned to the match list. The name can only contain the characters A-Z, a-z, 0-9, period (.), hyphen (-), and underline (_). The maximum length is 40 characters.

Table 7-100 (Cont.) Create Match List

Field Name	Description
Description	Free-form text
Type	Select from the following: • **String** (default) - The list consists of strings. • **Wildcard String** - The list consists of wildcard match patterns that use an asterisk (*) to match zero or more characters or a question mark (?) to match exactly one character. • **IPv4 Subnet** - IP address in IPv4 subnet. • **IPv6 Subnet** - IP address in IPv6 subnet. • **Regular Expression** - Regular Expression to match the IP address.
Items	Item list to either include or exclude.

4. Click **Save**.
   The match list gets listed on the **Match List** page. It is is defined in the database and can now be used in a policy.

> **Note**
>
> Use or available under the **Actions** column to update or delete the match list.

Importing the Match Lists

To import match lists:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.1.6 Schemas

Schemas enable operators to accept data in custom format instead of the standard format. This data can then be used to construct conditions and actions.

This section describes how to configure Schemas for Policy.

To access Schemas configurations from CNC Console home page, expand **Policy**, navigate to **Policy Data Configurations**, click **Common**, and select **Schemas**.

On clicking **Schemas**, you can select to customize any of the following configurations:

- Yaml Schemas
- Diameter Schemas

7.4.1.6.1 Yaml Schema

This procedure provides information about how to view, import, and configure the custom schemas.

The **Yaml Schema** page allows you to import and manage existing Yaml Schemas. The page displays the list of defined Yaml schemas and provides the options to import, export, update, or delete the schemas.

This section describes how to import a custom schema Yaml file in the CNC Console.

> **Note**
>
> Custom schema yaml file must follow Open API standards.

To import a custom Yaml schema:

1. From the navigation menu under **Schemas**, select **Yaml Schemas**.
   This opens the **Yaml Schemas** page.

   The page lists the details including schema name, Schema Tag, and Actions for the following previsioned schema yaml files:

   - `AmPolicyData`
   - `SmPolicyData`
   - `UePolicyData`

2. Click Import .
   The page opens the **File Upload** dialog box.
3. Upload the file in JSON format by using the **Drag and Drop** button.
4. Select the tags to tag the schemas in the selected files. The page supports only **User** tag.
5. Click **Import**.

On successful import, the yaml file gets listed on the **Yaml Schema** page.

> **Note**
>
> - To update any existing schema, click Export , make changes to the exported yaml file, and save it on your system. Then, import the file again.
> - To delete any existing service area restriction, click the under the **Action** column.

When a schema is tagged as **User** during import, CNC Policy allows you to access and configure the schema using the **User Attributes** blocks, available under the Public category. When a schema is not tagged, you may access and configure it using the **Custom Attributes** blocks, available under the Public category. For more information, see *Oracle Communications Cloud Native Core Policy Design Guide*.

Sample Yaml Schema

The folllowing is a sample yaml schema:

```
openapi: 3.0.0
info:
  description: Customer
  version: 0.0.1
  title: Customer
paths:
  /:
    get:
      operationId: get
      summary: get
      tags:
        - get
      responses:
        '200':
          description: OK
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Customer'
components:
  schemas:
    Customer:
      type: object
      properties:
        phones:
          type: array
          items:
             type: string
        name:
          type: string
        address:
          $ref: '#/components/schemas/Address'
    Address:
      type: object
      properties:
        house:
          type: string
        street:
          type: string
        city:
          type: string
```

7.4.1.6.2 Diameter Schemas

This section describes the diameter schemas for user data configurations.

To access Schemas configurations under **Schemas**, select **Diameter Schemas**.

On clicking **Diameter Schemas**, you can select to customize any of the following configurations:

- Custom AVP

- Custom Vendor

7.4.1.6.2.1 Custom AVP

An attribute-value pair (AVP) is used to encapsulate protocol specific information supported by CNC Policy. Diameter messages, such as RAA, CCA, CCR, and RAR are supported by third-party AVP policy conditions. The supported outgoing Diameter messages set or remove the third-party AVPs. For more information, see Custom AVP to Support Third-Party Vendor Specific AVPs.

To create a custom AVP:

1. From the navigation menu, under **Diameter Schemas**, select **Custom AVP**.
   This opens the **Custom AVP** page. The page lists the existing AVPs. You can add or import new AVPs using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Custom AVP** page.
3. Enter values for the input fields available on the page:
   The following table describes the fields:

Table 7-101 Create Custom AVP

Field Name	Description
AVP Name	The name you assign to the AVP. This is a mandatory field. The name can only contain the characters A–Z, a–z, 0–9, period (.), hyphen (-), and underline (_). The maximum length is 255 characters.
Description	Free-form text that identifies the AVP. Enter up to 250 characters.
AVP Code	A unique numeric value assigned to the new AVP. This is a mandatory field.
Vendor	Select a vendor from the vendor list. To add a vendor to the list, see Custom Vendor.
Mandatory Flag	
Protect Flag	When checked, specifies the protected AVP values.
May Encrypt Flag	The AVP is encrypted if the checkbox is specified.
Vendor Specific Flag	The AVP is encrypted if the checkbox is specified.

Table 7-101 (Cont.) Create Custom AVP

Field Name	Description
AVP Type	Select the data type from the list: • **address** • **enumerated** • **float32** • **float64** • **grouped** • **id** • **int32** • **int64** • **ipFilterRule** • **octetString** • **time** • **uint32** • **uint64** • **uri** • **utf8String**
Parent AVP	If the AVP is a member of a grouped AVP, then the parent AVP must be specified. Select one of the following from the list: • **ADC-Rule-Definition:10415** • **ADC-Rule-Install:10415** • **ADC-Rule-Remove:10415** • **ADC-Rule-Report:10415** • **AF-Correlation-Information:10415** • **Acceptable-Service-Info:10415** • **Access-Network-Charging-Identifier-Gx:10415** • **Access-Network-Charging-Identifier:10415** • **Access-Network-Physical-Access-ID:10415** • **Allocation-Retention-Priority:10415** • **Application-Detection-Information:10415** • **CC-Money** • **Charging-Information:10415** • **Charging-Rule-Definition-3GPP2:5535** • **Charging-Rule-Definition:10415** • **Charging-Rule-Event-Cisco:9** • **Charging-Rule-Event-Trigger-Cisco:9** • **Charging-Rule-Install-3GPP2:5535** • **Charging-Rule-Install:10415** • **Charging-Rule-Remove:10415** • **Charging-Rule-Report-3GPP2:5535** • **Charging-Rule-Report:10415** • **Codec-Data-Tmp:10415**

4. Click **Save**.
5. If the AVP name matches the name of a standard AVP, a confirmation message displays. Click **OK** to overwrite the existing AVP.

The Custom AVP gets listed on the **Custom AVP** page.

> **Note**
>
> Use  or  available under the **Actions** column to update or delete the AVP.

Importing Custom AVP

To import the custom AVP:

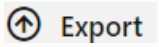
1. Click  .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.1.6.2.2 Custom Vendor

A custom vendor is used to define a vendor in the the PCRF system. This dictionary includes vendor IDs and text descriptions. You can define custom vendors and add them to the dictionary.

To configure Custom Vendor:

1. From the navigation menu, under **Diameter Schemas**, select **Custom Vendor**.
   This opens the **Custom Vendor** page. The page lists the existing vendors. You can add or import new vendors using this page.

   > **Note**
   >
   > Click  to download the available listings in the JSON file format on your system.

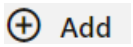
2. Click  .
   This opens the **Create Custom Vendor** page.
3. Enter values for the input fields available on the page:
   The following table describes the fields:

Table 7-102 Create Custom Vendor

Field Name	Description
Vendor Name	The name you assign to the vendor. This is a mandatory field. The name can only contain the characters A–Z, a–z, 0–9, period (.), hyphen (-), and underline (_). The maximum length is 255 characters.
Description	Free-form text that identifies the vendor. Enter up to 250 characters.
Vendor Code	A unique numeric value assigned to the new vendor. This is a mandatory field.

4. Click **Save**.

The Custom Vendor gets listed on the **Custom Vendor** page.

> **Note**
>
> Use or available under the **Actions** column to update or delete the vendor.

Importing Custom Vendor

To import custom vendor:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.2 PCF Session Management

This section includes the configurations for PCF Session Management service.

To access PCF Session Management configurations from the CNC Console home page, expand **Policy**, navigate to **Policy Data Configurations** and select **PCF Session Management**.

The PCF Session Management configurations includes:

- Session Rule
- Session Rule Profile
- QoS Information
- PCC Rule
- PCC Rule Profile
- QoS Data
- Charging Data
- Traffic Control Data
- Condition Data

7.4.2.1 Session Rule

This procedure provides information about how to create and manage the session rules in CNC Policy.

The **Session Rule** page allows you to create new and manage existing session rule configurations. The page displays the list of defined rules and provides the options to import, export, or add data.

To configure the session rules:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **PCF Session Management**, and then, select **Session Rule**.
   This opens the **Session Rule** page. The page lists the existing session rules data. You can add or import new rules using this page.

> **Note**
>
> Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Session Rule** page.
3. On the **Create Session Rule** page, enter the information common to all the groups available on the page.
   The following table describes the common input fields:

Table 7-103 Create Session Rule

Field Name	Description
Session Rule ID	Specifies the Session Rule ID.
Name	Specifies the name assigned to the session rule.
Description	Free-form text that identifies the session rule.

4. Expand the **Authorized Session AMBR** group.
   This group allows you to add authorized session AMBR information.
5. Enter the values for the input fields, available under the **Authorized Session AMBR** group.
   The following table describes the fields:

Table 7-104 Authorized Session AMBR

Field Name	Description
Uplink Bandwidth	Specifies the bandwidth in uplink.
Downlink Bandwidth	Specifies the bandwidth in downlink.

> **Note**
>
> Click the **Remove** button to remove the Authorized Session AMBR details from the page.

6. Select value for **Condition Data** from the drop-down list.
7. Select value for **Authorize Default Qos** from the drop-down list.

> **Note**
>
> The drop-down list data is configured from the QoS Information. For more information about QoS Data configuration, see QoS Data.

8. Click **Save**.
   The session rule data gets listed on the **Session Rule** page.

> **Note**
>
> Use or available under the **Actions** column to update or delete the session rule data configurations.

Importing Session Rules

To import session rules:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.2.2 Session Rule Profile

This procedure provides information about how to create and manage the session rule profiles that can be assigned to session rule profiles.

The **Session Rule Profile** page allows you to create new and manage existing session rule profile configurations. The page displays the list of defined rules and provides the options to import, export, or add data.

To configure the session rule profiles:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **PCF Session Management**, and then, select **Session Rule Profile**.
   This opens the **Session Rule Profile** page. The page lists the existing session rule profiles. You can add or import new profiles using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Session Rule Profile** page.
3. On the **Create Session Rule Profile** page, enter the information common to all the groups available on the page.
   The following table describes the common input fields:

Table 7-105 Create Session Rule Profile

Field Name	Description
Session Rule Profile NAME	Specifies the name assigned to the session rule profile.
Description	Free-form text that identifies the session rule profile.

4. Expand the **Authorized Session AMBR** group.
   This group allows you to add authorized session AMBR information.

5. Enter the values for the input fields, available under the **Authorized Session AMBR** group.
   The following table describes the fields:

Table 7-106 Authorized Session AMBR

Field Name	Description
Uplink Bandwidth	Specifies the bandwidth in uplink.
Downlink Bandwidth	Specifies the bandwidth in downlink.

> **Note**
>
> Click the **Remove** button to remove the Authorized Session AMBR details from the page.

6. Select value for **Condition Data** from the drop-down list.
7. Select value for **Authorize Default Qos** from the drop-down list.

> **Note**
>
> The drop-down list data is configured from the QoS Information. For more information about QoS Data configuration, see QoS Data.

8. Click **Save**.
   The profile data gets listed on the **Session Rule Profile** page.

> **Note**
>
> Use  or  available under the **Actions** column to update or delete the session rule profile configurations.

Importing the Session Rule Profiles

To import session rule profiles:

1. Click  .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.2.3 QoS Information

This procedure provides information about how to create and manage the Quality of Service (QoS) information in CNC Policy.

The **QoS Information** page allows you to create new and manage existing information related to QoS configurations. The page displays the list of defined QoS identifiers and provides the options to import, export, or add data.

To configure QoS Information data:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **PCF Session Management**, and then, select **QoS Information**.
   This opens the **QoS Information** page. The page lists the existing session QoS data. You can add or import new data using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create QoS Information** page.
3. On the **Create QoS Information** page, enter values for the input fields common to all the groups available on the page.
   The following table describes the fields:

Table 7-107 Common Configuration on Create QoS Information

Field Name	Description
Name	Specifies the name assigned to the QOS information.
Description	Free-form text that identifies the QOS information.
Default 5G QoS Identifier	Identifier for the authorized QoS parameters for the service data flow. It is included when the QoS information decision is initially provisioned.
Priority Level	Unsigned integer indicating the 5QI Priority Level, within a range of 1 to 127.
Average Window	Represents the duration over which the guaranteed and maximum bitrate is calculated (NOTE).
Max DataBurstVol	Denotes the largest amount of data that is required to be transferred within a period of 5GAN PDB (NOTE).

4. Expand the **ARP** group.
   This group allows you to add Allocation and Retention Priority (ARP) information.
5. Enter the values for the input fields, available under the **ARP** group.
   The following table describes the fields:

Field Name	Description
Priority Level	Unsigned integer indicating the ARP Priority Level, within the range 1 to 15.
Preemption Capability	Defines whether a service data flow may get resources that were already assigned to another service data flow with a lower priority level. Possible values are: • NOT_PREEMPT : Does not trigger pre-emption. • MAY_PREEMPT : May trigger pre-emption.

Field Name	Description
Preemption Vulnerability	Defines whether a service data flow may lose the resources assigned to it in order to admit a service data flow with higher priority level. Possible values are: • NOT_PREEMPTABLE : Does not be pre-empted. • PREEMPTABLE : May be pre-empted.

> **Note**
>
> Click the **Remove** button to remove the ARP fields from the page.

6. Click **Save**.
   The QoS information data gets listed on the **QoS Information** page.

> **Note**
>
> Use  or  available under the **Actions** column to update or delete the QoS information.

Importing the QoS Information

To import the QoS Information:

1. Click 

   .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.2.4 PCC Rule

This procedure provides information about how to create and manage the PCC Rules in Policy.

The **PCC Rule** page allows you to create new and manage existing PCC Rule configurations. The page displays the list of defined rules and provides the options to import, export, or add rules.

To configure the PCC rule:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **PCF Session Management**, and then, select **PCC Rule**.
   This opens the **PCC Rule** page. The page lists the existing session PCC rules data. You can add or import new rules using this page.

> **Note**
>
> Click 
>
> to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create PCC Rule** page.
3. On the **Create PCC Rule** page, enter values for the available input fields.
   The following table describes the fields:

Table 7-108 Create PCC Rule

Field Name	Description
PCC Rule Id	Specifies the PCC Rule ID.
Name	Specifies the name assigned to the PCC rule.
Description	Free-form text that identifies the PCC rule.
Type	Select the required type. Possible values are: • Predefined PCC Rule • Dynamic PCC Rule If you have selected Dynamic PCC Rule, then perform Step 4 else, perform Step 5.
Use Custom QoS Class Identifier	When this flag is enabled, you can use the custom QCI values. Defalut Value: Disabled
QoS Class Identifier (1-254)	When Use Custom QoS Class Identifier flag is enabled, you can use a custom value for this parameter. The value ranges between 1 and 254. When Use Custom QoS Class Identifier flag is disabled, you can select one of the QCIs from the dropdown list box: • 4=Streaming • 5-Interactive with Priority 1 Signalling • 6=Interactive with Priority 1 • 7=Interactive with Priority 2 • 8=Interactive with Priority 3 • 9=Background • 65=MC-PTT Voice • 66-PTT Voice

4. Expand the **Flow Information** group.
   The expanded group displays the available flow information and allows you to add new information.

   To add flow information:

   a. Click Add .
      The page opens the **Add Flow Information** dialog box.
   b. On the dialog box, enter values for the following input fields:

Table 7-109 Add Flow Information

Field Name	Description
Name	Indicates the name for the flow.
Pack Filt Id	An identifier of packet filter.

Table 7-109 (Cont.) Add Flow Information

Field Name	Description
Packet Filter Usage	The packet is sent to the UE. The default value "FALSE" applies, if the attribute is not present and has not been supplied previously.
Tos Traffic Class	Contains the IPv4 Type-of-Service and mask field or the IPv6 Traffic-Class field and mask field.
Spi	The security parameter index of the IPSec packet.
Flow Label	The IPv6 flow label header field.

c. Click **Save**.
The Flow Information gets listed under the **Flow Information** group on the **Create PCC Rule** page.

5. On the **Create PCC Rule** page, enter the values for the following input fields:

Table 7-110 Create PCC Profile Common Configurations

Field Name	Description
App Id	A reference to the application detection filter configured at the UPF.
Content Version	Indicates the content version of the PCC rule.
Precedence	Determines the order in which this PCC rule is applied relative to other PCC rules within the same PDU session. It is included if the "flowInfos" attribute is included or may be included if the "appId" attribute is included when the PCF initially provisions the PCC rule.
AF Signalling Protocol	Indicates the protocol used for signaling between the UE and the AF. The default value "NO_INFORMATION" applies, if the attribute is not present and has not been supplied previously.
--QoS Flow Usage	Indicates the QoS flow usage. The default value is not applicable for this field. Users can select GENERAL or IMS_SIG from the drop down list as per the requirement.
Application Relocation	Indication of application relocation possibility. The default value "NO_INFORMATION" is applicable, if the attribute is not present and has not been supplied previously.
Qos Data	A reference to the QoSData policy type decision type.
Traffic Control Data	A reference to the TrafficControlData policy decision type.
Charging Data	A reference to the ChargingData policy decision type.
Usage Monitoring Data	A reference to UsageMonitoringData policy decision type.
Condition Data	A reference to the condition data.

6. On the **Create PCC Rule** page, click **Save**.
The PCC profile data gets listed on the **PCC Rule** page.

> **Note**
>
> Use or available under the **Actions** column to update or delete the rule data.

Importing the PCC Rules

To import the pcc rules:

1. Click **Import**.
   The **File Upload** window appears on the screen.
2. Upload the files in required format by clicking **Drop Files here or click to upload**.

7.4.2.5 PCC Rule Profile

This procedure provides information about how to create and manage the PCC Rule Profiles in Policy.

The **PCC Rule Profile** page allows you to create new and manage existing PCC Rule Profile configurations. The page displays the list of defined profiles and provides the options to import, export, or add profiles.

To configure PCC Rule Profile:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **PCF Session Management**, and then, select **PCC Rule Profile**.
   This opens the **PCC Rule Profile** page. The page lists the existing session PCC rule profile data. You can add or import new profiles using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create PCC Rule Profile** page.
3. On the **Create PCC Rule Profile** page, enter values for the available input fields.
   The following table describes the fields:

Table 7-111 Create PCC Rule Profile

Field Name	Description
Name	Specifies the name assigned to the PCC rule profile.
Description	Free-form text that identifies the PCC rule profile.
Type	Select the required type. Possible Values are: • Predefined PCC Rule • Dynamic PCC Rule If you have selected Dynamic PCC Rule, then perform Step 4 else, perform Step 5.

Table 7-111 (Cont.) Create PCC Rule Profile

Field Name	Description
Use Custom QoS Class Identifier	When this flag is enabled, you can use the custom QCI values. Defalut Value: Disabled
QoS Class Identifier (1-254)	When Use Custom QoS Class Identifier flag is enabled, you can use a custom value for this parameter. The value ranges between 1 and 254. When Use Custom QoS Class Identifier flag is disabled, you can select one of the QCIs from the dropdown list box: • 4=Streaming • 5-Interactive with Priority 1 Signalling • 6=Interactive with Priority 1 • 7=Interactive with Priority 2 • 8=Interactive with Priority 3 • 9=Background • 65=MC-PTT Voice • 66-PTT Voice

4. Expand the **Flow Information** group.
The expanded group displays the available flow information and allows you to add new information.

   To add flow information:

   a. Click ⊕ Add .
   The page opens the **Add Flow Information** dialog box.

   b. On the dialog box, enter values for the following input fields:

   Table 7-112 Add Flow Information

Field Name	Description
Name	Indicates the name for the flow.
Pack Filt Id	An identifier of packet filter.
Packet Filter Usage	The packet is sent to the UE. The default value "FALSE" is applicable, if the attribute is not present and has not been supplied previously.
Tos Traffic Class	Contains the IPv4 Type-of-Service and mask field or the IPv6 Traffic-Class field and mask field.
Spi	The security parameter index of the IPSec packet.
Flow Label	The IPv6 flow label header field.

   c. Click **Save**.
   The Flow Information gets listed under the **Flow Information** group on the **Create PCC Rule Profile** page.

5. On the **Create PCC Rule Profile** page, enter the values for the following input fields:

Table 7-113 Create PCC Rule Profile Common Configurations

Field Name	Description
App Id	A reference to the application detection filter configured at the UPF.
Content Version	Indicates the content version of the PCC rule.
Precedence	Determines the order in which this PCC rule is applied relative to other PCC rules within the same PDU session. It is included if the "flowInfos" attribute is included or may be included if the "appId" attribute is included when the PCF initially provisions the PCC rule.
AF Signalling Protocol	Indicates the protocol used for signaling between the UE and the AF. The default value "NO_INFORMATION" is applicable, if the attribute is not present and has not been supplied previously.
--QoS Flow Usage	Indicates the QoS flow usage. The default value is not applicable for this field. Users can select GENERAL or IMS_SIG from the drop down list as per the requirement.
Application Relocation	Indication of application relocation possibility. The default value "NO_INFORMATION" is applicable, if the attribute is not present and has not been supplied previously.
Qos Data	A reference to the QoSData policy type decision type.
Traffic Control Data	A reference to the TrafficControlData policy decision type.
Charging Data	A reference to the ChargingData policy decision type.
Usage Monitoring Data	A reference to UsageMonitoringData policy decision type.
Condition Data	A reference to the condition data.

6. On the **Create PCC Rule Profile** page, click **Save**.
   The PCC profile data gets listed on the **PCC Rule Profile** page.

> **Note**
>
> Use  or  available under the **Actions** column to update or delete the profile data.

Importing PCC Rule Profiles

To import PCC Rule Profiles:

1. Click 
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.2.6 QoS Data

This procedure provides information about how to create and manage the QoS data in Policy.

The **QoS Data** page allows you to create new and manage existing data related to QoS configurations. The page displays the list of defined QoS details and provides the options to import, export, or add data.

To configure QoS data:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **PCF Session Management**, and then, select **QoS Data**.
   This opens the **QoS Data** page. The page lists the existing session QoS data. You can add or import new data using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create QoS Data** page.
3. On the **Create QoS Data** page, enter values for the input fields common to all the groups available on the page.
   The following table describes the fields:

Table 7-114 Create QoS Data Common Configurations

Field Name	Description
QoS ID	Specifies the QoS ID.
Name	Specifies the name assigned to the QOS data.
Description	Free-form text that identifies the QOS data.
Default 5G QoS Identifier	Identifier for the authorized QoS parameters of the service data flow. It is included when the QoS data decision is initially provisioned.
Maximum Bit Rate UL	Indicates the max bandwidth in uplink.
Maximum Bit Rate DL	Indicates the max bandwidth in downlink.
Guaranteed Bit Rate UL	Indicates the guaranteed bandwidth in uplink
Guaranteed Bit Rate DL	Indicates the guaranteed bandwidth in downlink.
QoS Notification Control	
Reflective QoS	Indicates whether the QoS information is reflective for the corresponding service data flow. Default value is "FALSE", if not present and has not been supplied previously.
Sharing Key UI	Indicates, by containing the same value, what PCC rules may share resource in uplink direction.
Sharing Key DI	Indicates, by containing the same value, what PCC rules may share resource in downlink direction.
Priority Level	Defines the relative importance of a resource request.
Averaging Window	Represents the duration over which the guaranteed and maximum bitrate is calculated (NOTE).
Maximum Data Burst Volume	Denotes the largest amount of data that is required to be transferred within a period of 5GAN PDB (NOTE).

Table 7-114 (Cont.) Create QoS Data Common Configurations

Field Name	Description
Maximum Packet Loss Rate Dl	Indicates the uplink maximum rate for lost packets that can be tolerated for the service data flow.
Max Packet Loss Rate Ul	Indicates the uplink maximum rate for lost packets that can be tolerated for the service data flow.
Default QoS Flow Indication	Indicates that the dynamic PCC rule always have its binding with the QoS Flow associated with the default QoS rule. Default value is "FALSE", if not present and has not been supplied previously.

4. Expand the **ARP** group.
   This group allows you to add Allocation and Retention Priority (ARP) information.
5. Enter the values for the input fields, available under the **ARP** group.
   The following table describes the fields:

Table 7-115 ARP Configurations

Field Name	Description
Priority Level	Defines the relative importance of a resource request.
Preemption Capability	Defines whether a service data flow may get resources that were already assigned to another service data flow with a lower priority level. Possible values are: • NOT_PREEMPT • MAY_PREEMPT
Preemption Vulnerability	Defines whether a service data flow may lose the resources assigned to it in order to admit a service data flow with higher priority level. Possible values are: • NOT_PREEMPTABLE • PREEMPTABLE

6. Click **Save**.
   The QoS data gets listed on the **QoS Data** page.

> **Note**
>
> Use  or  available under the **Actions** column to update or delete theQoS data.

Importing QoS Data

To import QoS data:

1. Click  .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.

3. Click **Import**.

7.4.2.7 Charging Data

This procedure provides information about how to create and manage the charging data in Policy.

The **Charging Data** page allows you to create new and manage existing charging data configurations. The page displays the list of defined charging details and provides the options to import, export, or add data.

To configure charging data:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **PCF Session Management**, and then, select **Charging Data**.
   This opens the **Charging Data** page. The page lists the existing charging data. You can add or import new data using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Charging Data** page.
3. On the **Create Charging Data** page, enter values for the following input fields:
   :

Table 7-116 Create Charging Data Configurations

Field Name	Description
Charging id	Specifies the charging ID
Name	A unique of the Charging data
Description	Brief description of the Charging data
Metering Method	The following options are available • DURATION • VOLUME • DURATION_VOLUME • EVENT Defines. which parameters must be metered for offline charging. If the attribute is not present, but it has been supplied previously, the previous information remains valid. If the attribute is not present and it has not been supplied previously or the attribute has been supplied previously but it is set to NULL, the metering method preconfigured at the SMF is applicable as the default metering method.

Table 7-116 (Cont.) Create Charging Data Configurations

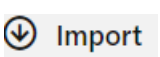
Field Name	Description
Offline	Indicates that offline charging is applicable to the PDU session or PCC rule. The default value is `false`, if the attribute is not present and it has not been supplied previously.
Online	Indicates that online charging is applicable to the PDU session or PCC rule. The default value is `false`, if the attribute is not present and it has not been supplied previously.
Rating Group	The charging key for the PCC rule used for rating purposes.
Reporting Level	The following options are available: • SER_ID_LEVEL • RAT_GR_LEVEL • SPON_CON_LEVEL Defines, on which level the SMF reports the usage for the related PCC rule. If the attribute is not present but it has been provided previously, the previous information remains valid. If the attribute is not present and it has not been supplied previously or the attribute has been supplied previously but it is set to NULL, the reporting level preconfigured at the SMF is applicable as the default reporting level.
Service Id	Indicates the identifier of the service or the service component in the service data flow of a PCC rule.
Sponsor Id	Indicates the sponsor identity.
App Svc Prov Id	Indicates the application service provider identity.
Af Charging Identifier	Identifies the charging control policy data within a PDU session.
sdfHandl	`sdfHandl` attribute send to SMF to indicate whether the service data flow is allowed to start, while the SMF is waiting for the response of the credit request from CHF.

4. Click **Save**.
The Charging data gets listed on the **Charging Data** page.

> **Note**
>
> Use  or  available under the **Actions** column to update or delete the charging data.

Importing Charging Data

To import charging data:

1. Click  .

The page opens the **File Upload** dialog box.

2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.2.8 Traffic Control Data

This procedure provides information about how to create and manage the Traffic Control data in Policy.

The **Traffic Control Data** page allows you to create new and manage existing traffic control configurations. The page displays the list of defined traffic control configurations and provides the options to import, export, or add data.

To configure traffic control data:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **PCF Session Management**, and then, select **Traffic Control Data**.
   This opens the **Traffic Control Data** page. The page lists the existing traffic control data. You can add or import new data using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Traffic Control Data** page.
3. On the **Create Traffic Control Data** page, enter the information common to all the groups available on the page.
   The following table describes the common input fields:

Table 7-117 Create Traffic Control Data Common Input Fields

Field Name	Description
Traffic Control id	Specifies the traffic control policy data id
Name	The name of the Traffic Control policy data
Description	The description of the Traffic Control policy data
Flow Status	The following options are available: • ENABLED-UPLINK • ENABLED-DOWNLINK • ENABLED • DISABLED • REMOVED Enum determining what action to perform on traffic. Possible values are: [enable, disable, enable_uplink, enable_downlink] . The default value "ENABLED" is applicable, if the attribute is not present and has not been supplied previously.

4. Expand the **Redirect Information** group.

This group allows you to add traffic redirect information.

5. Enter the values for the input fields, available under **Redirect Information** group.
The following table describes the fields:

Table 7-118 Redirect Information

Field Name	Description
Redirect Enabled	Indicates the redirect is enabled. This is an optional field.
Redirect Address Type	This string provides forward-compatibility with future extensions to the enumeration but is not used to encode content defined in the present version of this API.
Redirect Server Address	Indicates the address of the redirect server.

> **Note**
>
> Click the **Remove** button to remove the Redirect Information details from the page.

6. Enter the values for the following input fields:

Table 7-119 Create Traffic Control Data Common Input Fields

Field Name	Description
Mute Notification	Indicates whether application's start or stop notification is to be muted. The default value "FALSE" is applicable, if the attribute is not present and has not been supplied previously.
Traffic Steering Pol Id Dl	Reference to a preconfigured traffic steering policy for downlink traffic at the SMF.
Traffic Steering Pol Id Ul	Reference to a preconfigured traffic steering policy for uplink traffic at the SMF.

7. Expand the **Route To Locs** group.
The expanded group displays the available routes and allows you to create new routes.

   To create new routes:

   a. Click ⊕ Add .
   The page opens the **Add Route to Locs** dialog box.

   b. Enter the value for the **DNAI**.
   The DNAI value identifies the location of the application.

   c. Expand the **Route Information** group.
   This group allows you to add the traffic routing information.

   d. Enter the values for the input fields, available under **Route Information** group.
   The following table describes the fields:

Table 7-120 Redirect Information

Field Name	Description
Ipv6 Addr	Ipv6 address of the tunnel end point in the data network.
Port Number	UDP port number of the tunnel end point in the data network.
Ipv4 Addr	Ipv4 address of the tunnel end point in the data network.

> **Note**
>
> Click the **Remove** button to remove the Route Information details from the dialog box.

e. Enter the value for **Route Profile Id**.
This value identifies the routing profile Id.

f. Click **Save**.
The Route Information gets listed under the **Route to Locs** group on the **Create Traffic Control Data** page.

8. On the **Create Traffic Control Data** page, expand the **Up Path Chg Event** group.

9. Enter the values for the input fields, available under **Up Path Chg Event** group.
The following table describes the fields:

Table 7-121 Up Path Chg Event

Field Name	Description
Notification Uri	Defines the notification Uri sent by the SMF.
Notification Correlation Id	It is used to set the value of Notification Correlation ID in the notification sent by the SMF.
Dnai Change Type	The following options are available: • EARLY • EARLY_LATE • LATE Possible values are EARLY: Early notification of UP path reconfiguration. - EARLY_LATE: Early and late notification of UP path reconfiguration. This value is present in the subscription to the DNAI change event. LATE: Late notification of UP path reconfiguration. This string provides forwardcompatibility with future extensions to the enumeration but is not used to encode content defined in the present version of this API.

> **Note**
>
> Click the **Remove** button to remove the Redirect Information details from the page.

10. On the **Create Traffic Control Data** page, click **Save**.
    The traffic control data gets listed on the **Traffic Control Data** page.

> **Note**
>
> Use or available under the **Actions** column to update or delete the traffic control data configurations.

Importing Traffic Control Data

To import the traffic control data:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.2.9 Condition Data

This procedure provides information about how to create and manage the Condition data in Policy.

The **Condition Data** page allows you to create new and manage existing conditions that can be applied for Policy services. The page displays the list of defined conditions and provides the options to import, export, or add data.

To configure condition data:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **PCF Session Management**, and then, select **Condition Data**.
   This opens the **Condition Data** page. The page lists the existing conditions. You can add or import new data using this page.

> **Note**
>
> Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Condition Data** page.
3. On the **Create Condition Data** page, enter values for the following input fields:
   :

Table 7-122 Create Condition Data Configurations

Field Name	Description
Condition id	Specifies the condition data policy data id.
Name	The name of the Condition Data policy data.
Description	The description of the Condition Data policy data.
Activation Time	The time when the decision data is activated.
Deactivation Time	The time when the decision data is deactivated.

4. Click **Save**.
   The condition gets listed on the **Condition Data** page.

> **Note**
>
> Use or available under the **Actions** column to update or delete the condition data.

Importing Condition Data

To import condition data:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.3 PCF Access and Mobility

This section includes the configurations for PCF Access and Mobility service.

To access PCF Access and Mobility configurations from the CNC Console home page, expand **Policy**, navigate to **Policy Data Configurations** and select **PCF Access and Mobility**.

The PCF Access and Mobility configuration includes Managing Service Area Restriction.

7.4.3.1 Service Area Restriction

This procedure provides information about how to create and manage the Service Area Restriction (SAR) in Policy.

The **Service Area Restriction** page allows you to create new and manage existing restrictions. The page displays the list of defined rules and provides the options to import, export, or add data.

To configure service area restrictions:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **PCF Access and Mobility**, and then, select **Service Area Restriction**.
   This opens the **Service Area Restriction** page. The page lists the details of existing service area restrictions including their name and restriction type in a tabular format. You can add or import new data using this page.

> **Note**
>
> Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Service Area Restriction** page.
3. On the **Create Service Area Restriction** page, enter the information common to all the groups available on the page.
   The following table describes the common input fields:

Table 7-123 Create Service Area Restriction Configurations

Field Name	Description
Name	Use this field to add a customized name for service area restriction.
Description	Use this field to add a customized description for service area restriction.
Restriction Type	Use this field to specify the restriction type. Possible values are: • ALLOWED_AREAS • NOT_ALLOWED_AREAS **Note:** Set a restriction type for service area restriction only when **Areas** attribute is available.

4. Expand the **Areas** group.
   This group lists the existing area details and allows you to configure new areas.
5. To add new areas:
   a. Click Add .
      The page opens the **Add Areas** dialog box.
   b. Enter the value for the following fields:

Table 7-124 Add Areas Configurations

Field Name	Description
Tacs	The value for Type Allocation Codes (Tacs) can be defined as a hexa decimal number between 0 and 65535.
Area Codes	Enter the value of area codes.

   c. Click **Save**.
      The areas gets listed under the **Areas** group on the **Create Service Area Restriction** page.
6. Enter value for the **Max Number of TAs** field.
7. Click **Save**.
   The SAR data gets listed on the **Service Area Restriction** page.

> **Note**
>
> Use  or  available under the **Actions** column to update or delete the SAR data.

Importing Service Area Restrictions

To import the Service Area Restrictions:

1. Click  .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.4 PCF UE Policy

This section includes the configurations for PCF UE Policy service.

To access PCF UE Policy configurations from the CNC Console home page, expand **Policy**, navigate to **Policy Data Configurations** and select **PCF UE Policy**.

The PCF UE Policy configurations includes:

- URSP Rule
- UPSI Rule

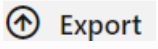
7.4.4.1 URSP Rule

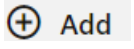
This procedure provides information about how to create and manage the UE Route Selection Policy (URSP) Rules in Policy.

The **URSP Rule** page allows you to create new and manage existing reules. The page displays the list of defined URSP rule configurations and provides the options to import, export, or add rules.

To configure URSP Rules:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **PCF UE Policy**, and then, select **URSP Rule**.
   This opens the **URSP Rule** page. The page lists the existing URSP rules data. You can add or import new rules using this page.

   > **Note**
   >
   > Click  to download the available listings in the JSON file format on your system.

2. Click  .
   This opens the **Create URSP Rule** page.
3. On the **Create URSP Rule** page, enter the information common to all the groups available on the page.

The following table describes the common input fields:

Table 7-125 Create URSP Rule Configurations

Field Name	Description
Name	Name of the URSP rule.
Precedence	Precedence value of the URSP rule.

> **Note**
>
> Name can not be the substring of any other configured URSP rule Name.

4. Expand the **Traffic Descriptor** group.
The expanded group allows you to create new descriptors.

   To create new traffic descriptors:

   a. Click Add .
   The page opens the **Add Traffic Descriptor** dialog box.

   b. Select a value from the **Type** drop-down list and then, enter corresponding values, if applicable. The following table lists the traffic descriptor types and their corresponding values:

Table 7-126 Traffic Descriptor Type and Values

Traffic Descriptor Type	Value
MATCH_ALL	NA
OS_ID_OS_APP_ID	• OS ID • OS APP ID
IPV4_REMOTE_ADDRESS	• IPv4 Address • Subnet Mask
IPV6_REMOTE_ADDRESS	• IPv6 Address • IPv6 Prefix Length
PROTOCOL_IDENTIFIER	Protocol Number
SINGLE_REMOTE_PORT	Remote Port
REMOTE_PORT_RANGE	• Start Port • End Port
SECURITY_PARAMETER_INDEX	IPSec Security Parameter Index
TYPE_OF_SERVICE_CLASS	• Type of Service • Mask
FLOW_LABEL	IPv6 Flow Label
DESTINATION_MAC_ADDRESS	MAC Address
T_802_1Q_C_TAG_VID	Customer VLAN ID
T_802_1Q_S_TAG_VID	Service VLAN ID
T_802_1Q_C_TAG_PCP_DEI	• Priority Code Point (PCP) • Drop Eligible Indicator (DEI)
T_802_1Q_S_TAG_PCP_DEI	• Priority Code Point (PCP) • Drop Eligible Indicator (DEI)
ETHERTYPE	Ethertype

Table 7-126 (Cont.) Traffic Descriptor Type and Values

Traffic Descriptor Type	Value
DNN	DNN
CONNECTION_CAPABILITIES	Connection Capabilities
DESTINATION_FQDN	FQDN
OS_APP_ID	OS APP ID

c. Click **Save**.
The Route Information gets listed under the **Traffic Descriptor** group on the **Create URSP Rule** page.

5. Expand the **Route Selection Descriptor List** group.
The expanded group allows you to create new descriptors.

To create new route selection descriptors:

a. Click Add .
The page opens the **Add Route Selection Descriptor List** dialog box.

b. Enter the value in the **Precedence** field.

c. Expand **Route Selection Descriptor Components** group.
The expanded group allows you to create new descriptors components.

d. To add new components:

i. Click Add .
The page opens the **Add Route Selection Descriptor Components** dialog box.

ii. Select a value from the **Type** drop-down list and then, enter corresponding values, if applicable. The following table lists the route selection descriptor component types and their corresponding values:

Table 7-127 Route Selection Descriptor Components Configurations

Type	Value
SSC_MODE	Select any of the following values for SSC Mode: • SSC Mode 1 • SSC Mode 2 • SSC Mode 3
SNSSAI	Enter the values for the following fields: • SST • SD • Mapped SST • Mapped SD
DNN	Enter the value of DNN in the provided field.
PDU_SESSION_TYPE	Select any of the following values for PDU Session Type: • IPV4 • IPV6 • IPV4V6 • Unstructured • Ethernet

Table 7-127 (Cont.) Route Selection Descriptor Components Configurations

Type	Value
PREFERRED_ACCESS_TYPE	Select any of the following values for Preferred Access Type: • 3GPP Access • Non 3GPP Access
NON_SEAMLESS_NON_3GPP_OFFLOAD_INDICATION	NA

iii. Click **Save** on the **Add Route Selection Descriptor Components** dialog box.

e. Click **Save** on the on the **Add Route Selection Descriptor Lists** dialog box.

6. On the **Create URSP Rule** page, click **Save**.
The URSP rule data gets listed on the **URSP Rule** page.

> **Note**
>
> Use or available under the **Actions** column to update or delete the URSP rule configurations.

Importing URSP Rule

To import the URSP rules:

1. Click Import .
The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.4.2 UPSI

This procedure provides information about how to create and manage the UE Policy Section Identifier (UPSI) in Policy.

The **UPSI** page allows you to create new and manage existing UPSI configurations. The page displays the list of defined configurations and provides the options to import, export, or add data.

To configure UPSI:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **PCF UE Policy**, and then, select **UPSI**.
This opens the **UPSI** page. The page lists the existing UPSI data. You can add or import new data using this page.

> **Note**
>
> Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create UPSI** page.
3. On the **Create UPSI** page, enter values for the input fields common to all the groups available on the page.
   The following table describes the fields:

Table 7-128 Create UPSI Configurations

Field Name	Description
Name	Name of the UPSI.
UPSC	Defines UE Policy Section Code. Enter a number between 0 and 65,535.
URSP Rules	Defines URSP rules.

> **Note**
>
> Name can not be the substring of any other configured URSP rule Name.

4. Expand the **PLMN** group.
   This group allows you to add PLMN information.

   The following table describes the fields:

Table 7-129 PLMN Configurations

Field Name	Description
MCC	Defines the Mobile Country Code. Enter a number between 0 and 999.
MNC	Defines the Mobile Network Code. Enter a number between 0 and 999.

5. Click **Save**.
   The UPSI data gets listed on the **UPSI** page.

> **Note**
>
> Use or available under the **Actions** column to update or delete the session UPSI configurations.

Importing UPSI

To import UPSIs:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.5 PCRF Core

This section includes the configurations for PCRF Core.

To access PCRF Core configurations from the CNC Console home page, expand **Policy**, navigate to **Policy Data Configurations** and select **PCRF Core**.

The PCRF Core configuration includes the following:

- Charging Server
- Media Profile
- Presence Reporting Area
- Time Periods
- Retry Profile
- Traffic Profile

7.4.5.1 Charging Server

This procedure provides information about how to define and manage charging servers within the PCRF Core in Policy. A charging server is an application that calculates billing charges.

The **Charging Server** page allows you to create new and manage existing charging server configurations. The page displays the list of defined configurations and provides the options to import, export, or add data.

To define a charging server:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **PCRF Core**, and then, select **Charging Server**.

   This opens the **Charging Server** page. The page lists the existing charging server data. You can add or import new data using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .

   This opens the **Create Charging Server** page.

3. On the **Create Charging Server** page, enter the following information in the available input fields:

Table 7-130 Create Charging Server Configurations

Field Name	Description
Name	The unique name you assign to the charging server. This is a mandatory field. The name can only contain the characters A–Z, a–z, 0–9, period (.), hyphen (-), and underline (_).

Table 7-130 (Cont.) Create Charging Server Configurations

Field Name	Description
Description/Location	Free-form text that identifies the charging server within the network. Enter up to 250 characters.
Host Name	The FQDN (fully qualified domain name assigned) to the charging server. This is a mandatory field.
Port	The port number on which the charging server is listening for messages. The default value is 3868.
Transport	Select the transport protocol used to communicate with the charging server. The available options include: • **tcp** Transmission Control Protocol (used with TACACS+) • **udp** User Datagram Protocol (used with RADIUS) **Note:** If you configure the Transport protocol as **udp**, you cannot configure the AAA Protocol as **diameter**.
Protocol	Select the Authentication, Authorization, and Accounting (AAA) Protocol used to communicate with the charging server. The available options include: • **diameter** • **radius**
Security	Select if the transport security is used to communicate with the charging server. By default, the slider option remains unselected.

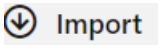
4. Click **Save**.

   The charging server gets listed on the **Charging Server** page.

> **Note**
>
> Use  or  available under the **Actions** column to update or delete the charging server configurations.

Importing Charging Server

To import charging server:

1. Click 

.

   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.5.2 Media Profile

This procedure provides information about how to define and manage media profiles under PCRF Core in Policy. A media profile describes a CODEC supported for Rx-to-PCMM translation in a cable network.

> **Note**
>
> Media Profiles is a function that is applicable to Cable mode only.

The **Media Profile** page allows you to create new and manage existing media profile configurations. The page displays the list of defined configurations and provides the options to import, export, or add data.

To configure media profile:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **PCRF Core**, and then, select **Media Profile**.

   This opens the **Media Profile** page. The page lists the existing profiles. You can add or import new profiles using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .

   This opens the **Create Media Profile** page.

3. On the **Create Media Profile** page, enter the following information in the available input fields:

Table 7-131 Create Media Profile Configurations

Field Name	Description
ID	The unique ID assigned to the media profile. This is a mandatory field.
Name	Unique name assigned to the media profile. This is a mandatory field.
Description	Brief description of the media profile.
Codec Name	Unique media subtype assigned to the media profile. This is defined in the IANA MIME registration for the CODEC. Enter a string of up to 255 characters. This is a mandatory field.

Table 7-131 (Cont.) Create Media Profile Configurations

Field Name	Description
Transport Type	Transport type. Select any of the following values from the drop-down list: • **RTP/AVP** (default) — RTP audio-video profile. • **RTP/SAVP** — RTP secure audio-video profile. • **RTP/AVPF** — RTP extended audio-video profile with feedback.
Payload Number	The payload number. Valid payload numbers range from 0 through 127. Enter -1 to indicate an unknown payload number. **Note:** You cannot add a CODEC that is predefined with a payload number in the range of 0 to 96. This is a mandatory field.
Sample Rate (kHz)	The sampling rate of the CODEC in KHz. The valid range is an integer from 1 through 100 KHz. This is a mandatory field.
Frame Size in Milliseconds	The size of one audio frame in milliseconds. This is the length of time represented by one audio frame. A single RTP packet may contain multiple audio frames. The bitrate is calculated using the frame size in milliseconds, the frame size in bytes, and the packetization time. The valid range is 0 through 100 ms. This is a mandatory field.
Frame Size in Bytes	The size of one audio frame size in bytes. This is the size represented by one audio frame. A single RTP packet may contain multiple audio frames. The bitrate is calculated using the frame size in milliseconds, the frame size in bytes, and the packetization time. The valid range is 1 through 1,500 bytes. This is a mandatory field.
Packetization Time	The length of time, in milliseconds, represented by the media in a packet. The bitrate is calculated using the frame size in milliseconds, the frame size in bytes, and the packetization time. The valid range is 1 through 100. This is a mandatory field.
Always Use Default Ptime	Select to always use the default packetization time, ignoring the value received in the SDP message. By default the slider option remains unselected. This is a mandatory field.

4. Click **Save**.

   The media profile gets listed on the **Media Profile** page.

> **Note**
>
> Use  or  available under the **Actions** column to update or delete the profiles.

Importing Media Profile

To import media profile:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.5.3 Presence Reporting Area

This procedure provides information about how to define and manage Presence Reporting Area (PRA) under PCRF Core in Policy.

The **Presence Reporting Area** page allows you to create new and manage existing PRA configurations. The page displays the list of defined configurations and provides the options to import, export, or add data.

To configure PCRF PRA:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **PCRF Core**, and then, select **Presence Reporting Area**.
   This opens the **Presence Reporting Area** page. The page lists the existing PRAs. You can add or import new PRAs using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Presence Reporting Area** page.
3. On the **Create Presence Reporting Area** page, enter values for the input fields common to all the groups available on the page.
   The following table describes the fields:

Table 7-132 Create Presence Reporting Area Configurations

Field Name	Description
ID	The unique identifying number of the PRA list. The ID must be a number between 0 and 16777125. This field is present if the Area of Interest subscribed or reported is a Presence Reporting Area.
Name	The unique name assigned to the PRA.
Description	Description of the PRA.
Type	Select the PRA type from the drop-down list. The available options are: • predefined: Select this to create a core network pre-configured PRA. • ue_dedicated: Select this to create a ue dedicated PRA.

4. Expand the **PRA Items** group.
   The expanded group allows you to add PRA items.

5. To add new items:

   a. Click ⊕ Add .
   The page opens the **Add PRA Items** dialog box.

   b. Select a value from the **Type** drop-down list and then, enter corresponding values. The available options with value combination are:

   - **TAI**: Tracking Area Identity (MCC,MNC,TAC)
   - **RAI**: Routing Area Identity (MCC,MNC,LAC,RAC)
   - **Macro eNodeB**: (MCC,MNC,MENB)
   - **Home eNodeB**: (MCC,MNC,HENB)
   - **ECGI**: E-UTRAN Cell Global Identifier (MCC,MNC,ECI)
   - **SAI**: Service Area Identity (MCC,MNC,LAC,SAC)
   - **CGI:** Cell Global Identity (MCC,MNC,LAC,CI)

   The input fields differ for each **Type**, based on the components applicable for that PRA item type. The following table describes each field:

Table 7-133 PRA Items Components Based on Type

Field Name	Description
MNC *Applicable for All*	Defines the Mobile Network Code. It can be a 2- or 3-digit number.
MCC *Applicable for All*	Defines the Mobile Country Code. It should be a 3-digit number.
TAC *Applicable for All*	28-bit string identifying an E-UTRAN Cell Id as specified, in hexadecimal representation. Each character in the string takes a value of "0" to "9" or "A" to "F" and represents 4 bits. The most significant character representing the 4 most significant bits of the Cell Id appears first in the string, and the character representing the 4 least significant bit of the Cell Id appears last in the string. Pattern: '^[A-Fa-f0-9]{7}$' Example: An E-UTRAN Cell Id 0x5BD6007 is encoded as "5BD6007".
LAC *Applicable for Type: RAI/SAI/CGI*	16-bit number that forms part of the Local Area Identifier (LAI) which includes the Mobile Country Code (MCC0 the Mobile Network Code (MNC) and the Local area code. Pattern: A decimal number between 0 and 65535.
RAC *Applicable for Type: RAI*	The routing area code for calling and differing of various RAs. Pattern: A decimal number between 0 and 65535.
MENB *Applicable for Type: Macro eNodeB*	Master eNodeB. Pattern: A decimal number between 0 and 268435455.

Table 7-133 (Cont.) PRA Items Components Based on Type

Field Name	Description
HENB *Applicable for Type: Macro eNodeB*	Home eNodeB Pattern: A decimal number between 0 and 268435455.
ECI *Applicable for Type: ECGI*	E-UTRAN Cell Identifier Pattern: A decimal number between 0 and 268435455.
SAC *Applicable for Type: SAI*	Service area code Pattern: A decimal number between 0 and 65535.
CI *Applicable for Type: CGI*	Cell identification Pattern: A decimal number between 0 and 65535.

c. Click **Save** on the on the **Add PRA Items** dialog box.

6. On the **Create Presence Reporting Area** page, click **Save**.
The PRA gets listed on the **Presence Reporting Area** page.

> **Note**
>
> Use or available under the **Actions** column to update or delete the PRA configurations.

Importing Presence Reporting Area

To import PRA:

1. Click Import .
The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.5.4 Time Periods

This procedure provides information about how to define and manage Time Periods under PCRF Core in Policy.

The **Time Periods** page allows you to view, create, and configure time periods. You can define a library of time periods to specify in policy time-of-day conditions and associate the time periods with multiple policies. Each time period can have one or more times slots defined. A time slot can be:

- Specific time of day
- Different days of the week
- Different days of a month
- Specific years
- Specific day and time in a specific year
- Specific day and time in every year

For example a single time period can have following time slots defined:

- Every Monday at 2 o'clock
- On the last day of the month
- On every Valentines day
- On May 17, 2016
- The first three days of March, July, and September

To configure time period:

1. From the navigation menu under **Policy**, navigate to **Policy Data Configurations**, click **PCRF Core**, and then, select **Time Periods**.
   This opens the **Time Periods** page. The page lists the existing time periods. You can add or import new periods using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Time Period** page.
3. On the **Create Time Period** page, enter values for the input fields common to all the groups available on the page.
   The following table describes the fields:

   Table 7-134 Create Time Periods Configurations

Field Name	Description
Name	Name of the time period. The name can only contain the characters A–Z, a–z, 0–9, period (.), hyphen (-), and underline (_).
Description	A descriptive phrase.
Prcendence	A positive integer. The lower the number, the higher the precedence. If time periods overlap, the time period with the highest precedence (lowest number) applies.

4. Expand the **Time Slot** group.
   The expanded group allows you to add time slots.
5. To add time slots:
   a. Click Add .
      The page opens the **Add Time Slot** dialog box.
   b. Enter the values for the following input fields:

Table 7-135 Add Time Slot Configurations

Field Name	Description
Years	Number of years.
Months of year	Number of months of the year.
Days of Month	Days of the month.
Days of Week	Days of the week.
Start Time	Starting time.
End Time	End time.

> **Note**
>
> The fields in the time slot configuration must have certain value. It must not be left empty.

c. Click **Save** on the **Add Time Slot** dialog box.

6. On the **Create Time Period** page, click **Save**.
The time period gets listed on the **Time Period** page.

> **Note**
>
> Use or available under the **Actions** column to update or delete the time period configurations.

Importing Time Periods

To import time periods:

1. Click Import .
The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.5.5 Retry Profile

This section describes how to configure Retry Profiles for PCRF Core.

To access Retry Profile configurations from the CNC Console home page, expand **Policy**, navigate to **Policy Data Configurations**, click **PCRF Core**, and select **Retry Profile**.

The Retry Profile configuration includes the following:

- PCC Retry Profile
- ADC Retry Profile

7.4.5.5.1 PCC Retry Profile

This procedure provides information about how to define and manage PCC Retry Profile under PCRF Core in CNC Policy.

The **PCC Retry Profile** page allows you to create new and manage existing profiles. The page displays the list of defined configurations and provides the options to import, export, or add data.

To configure PCC Retry Profile:

1. From the navigation menu under **Retry Profile**, select **PCC Retry Profile**.
   This opens the **PCC Retry Profile** page. The page lists the existing PCC Retry Profiles. You can add or import new profiles using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create PCC Retry Profile** page.
3. On the **Create PCC Retry Profile** page, enter values for the available input fields.
   The following table describes the fields:

Table 7-136 Create PCC Retry Profile Configurations

Field Name	Description
Name	Enter the Name for the profile. The name can only contain the characters A–Z, a–z, 0–9, period (.), hyphen (-), and underline (_). The maximum length is 255 characters.
Description	Enter the Description/Location. Free-form text describing the profile.
Maximum Retry Attempt (per Retry Cycle)	The maximum number of retry attempts during a retry cycle in the range from 1 to 10. The default is 5.
Initial Retry Interval	Enter the Initial Retry Interval. The length of time to wait, in seconds, after a reported failure or the end of the Back Off Interval before retrying. Enter a value from 0 to 30 seconds. The default is 10 seconds.
Maximum Retry Interval	Enter the Maximum Retry Interval. The maximum wait, in seconds, after a reported failure or the end of the Back Off Interval before retrying during a retry cycle. Enter a value from 1 to 180 seconds. The default is 60 seconds.
Back Off Interval	Enter the Back Off Interval. The interval from 1 to 86400 seconds between successive retry cycles. The default is 300 seconds.
Maximum Retry Cycles	Enter the Maximum Retry Cycles. The number of retry cycles ranging from 1 to 4. The default value is 1 cycle.
Rule Filure Code	Rule Failure Code The upper box lists available rule failure codes. The lower box lists rule failure codes installed in the profile.

4. Click **Save**.
   The PCC Retry Profile gets listed on the **PCC Retry Profile** page.

> **Note**
>
> Use  or  available under the **Actions** column to update or delete the PCC Retry Profile.

Importing PCC Retry Profile

To import PCC Retry Profile:

1. Click  .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

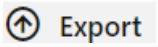
7.4.5.5.2 ADC Retry Profile

This procedure provides information about how to define and manage ADC Retry Profile under PCRF Core in CNC Policy.

The **ADC Retry Profile** page allows you to create new and manage existing profiles. The page displays the list of defined configurations and provides the options to import, export, or add data.

To configure ADC Retry Profile:

1. From the navigation menu under **Retry Profile**, select **ADC Retry Profile**.
   This opens the **ADC Retry Profile** page. The page lists the existing ADC Retry Profiles. You can add or import new profiles using this page.

> **Note**
>
> Click  to download the available listings in the JSON file format on your system.

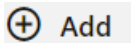
2. Click  .
   This opens the **Create ADC Retry Profile** page.
3. On the **Create ADC Retry Profile** page, enter values for the available input fields.
   The following table describes the fields:

Table 7-137 Create ADC Retry Profile Configurations

Field Name	Description
Name	Enter the Name for the profile. The name can only contain the characters A–Z, a–z, 0–9, period (.), hyphen (-), and underline (_). The maximum length is 255 characters.

Table 7-137 (Cont.) Create ADC Retry Profile Configurations

Field Name	Description
Description	Enter the Description/Location. Free-form text describing the profile.
Maximum Retry Attempt (per Retry Cycle)	The maximum number of retry attempts during a retry cycle in the range from 1 to 10. The default is 5.
Initial Retry Interval	Enter the Initial Retry Interval. The length of time to wait, in seconds, after a reported failure or the end of the Back Off Interval before retrying. Enter a value from 0 to 30 seconds. The default is 10 seconds.
Maximum Retry Interval	Enter the Maximum Retry Interval. The maximum wait, in seconds, after a reported failure or the end of the Back Off Interval before retrying during a retry cycle. Enter a value from 1 to 180 seconds. The default is 60 seconds.
Back Off Interval	Enter the Back Off Interval. The interval from 1 to 86400 seconds between successive retry cycles. The default is 300 seconds.
Maximum Retry Cycles	Enter the Maximum Retry Cycles. The number of retry cycles ranging from 1 to 4. The default value is 1 cycle.
Rule Filure Code	Rule Failure Code The upper box lists available rule failure codes. The lower box lists rule failure codes installed in the profile.

4. Click **Save**.
   The ADC Retry Profile gets listed on the **ADC Retry Profile** page.

> **Note**
>
> Use  or  available under the **Actions** column to update or delete the ADC Retry Profile.

Importing ADC Retry Profile

To import ADC Retry Profile:

1. Click  .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.5.6 Traffic Profile

This section describes how to configure Traffic Profiles for PCRF Core.

To access Retry Profile configurations from the CNC Console home page, expand **Policy**, navigate to **Policy Data Configurations**, click **PCRF Core**, and select **Traffic Profile**.

The Traffic Profile configuration includes the following:

- ADC Rule
- Diameter QoS
- PCC Profile
- PCC Rule
- Predefined ADC Rule
- Predefined ADC Rule Base
- Predefined PCC Rule
- Predefined PCC Rule Base

7.4.5.6.1 ADC Rule

This procedure provides information about how to define and manage ADC Rules under Traffic Profiles in PCRF Core.

The **ADC Rule** page allows you to create new and manage existing ADC Rules. The page displays the list of defined configurations and provides the options to import, export, or add data.

To confifure ADC Rule:

1. From the navigation menu under **Traffic Profile**, select **ADC Rule**.
   This opens the **ADC Rule** page. The page lists the existing ADC Rules. You can add or import new data using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .

   This opens the **Create ADC Rule** page.

3. On the **Create ADC Rule** page, enter values for the available input fields.
   The following table describes the fields:

Table 7-138 Create ADC Rule Configurations

Field Name	Description
Name	Name of the Profile.
Rule Name	Uniquely identifies the ADC rule. Used to reference an ADC rule in communication between the CNC Policy and a PCEF within one IP-CAN session.
Description	Description of the profile.
Uplink Max Authorized Rate (bps)	Maximum authorized bandwidth in bits per second for uplinks (user equipment to network).

Table 7-138 (Cont.) Create ADC Rule Configurations

Field Name	Description
Downlink Max Authorized Rate (bps)	Maximum authorized bandwidth in bits per second for downlinks (network to user equipment).
Monitoring Key	Monitoring key that may apply to the ADC rule.
Flow Status	Indicates whether the application traffic is enabled or disabled in either the uplink or downlink direction. Select from the following: • ENABLED_UPLINK • ENABLED_DOWNLINK • ENABLED • DISABLED
TDF Application Identifier	Identifies the traffic that belongs to the application to which the rule applies.
TDF Redirect Support	Indicates whether the application traffic should be redirected to another controlled address. Select from the following: • REDIRECTION_DISABLED • REDIRECTION_ENABLED
TDF Redirect Address Type	Specifies the format for the redirect address. Select from the following: • IPv4 • IPv6 • URL • SIP_URI
TDF Redirect Server Address	The address of the TDF redirect server in the specified address type.
Mute Notification	Notification to disable application detection notifications from the TDF device. Select **MUTE_REQUIRED** from the drop-down list. By default, mute remains disabled.
Service Identifier	Credit-control service identifier associated with the traffic defined by this rule. Only applicable if online charging is enabled.
Rating Group	Credit-control rating group associated with the traffic defined by this profile. Only applicable if online charging is enabled.
Reporting Level	Select from the following: • SERVICE_IDENTIFIER_LEVEL • RATING_GROUP_LEVEL
Online Charging	Specifies whether or not online charging is enabled in this profile. Select from the following: • DISABLE_ONLINE • ENABLE_ONLINE
Offline Charging	Specifies whether or not offline charging is enabled in this profile. Select from the following:: • DISABLE_OFFLINE • ENABLE_OFFLINE

Table 7-138 (Cont.) Create ADC Rule Configurations

Field Name	Description
Metering Method	Defines how service data-flow traffic is metered for offline charging. Select from the following: • DURATION • VOLUME • DURATION_VOLUME • EVENT
Precedence	Precedence value of the profile. The lower the precedence, the higher the priority.

4. Click **Save** to save the changes.

> **Note**
>
> Click **Cancel** to discard the changes.

The value gets listed on the **ADC Rule** page. Use or available in the next column to update or delete the listing.

Importing ADC Rule

To import ADC Rule:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.5.6.2 Diameter QoS

This procedure provides information about how to define and manage Diameter QoSs under Traffic Profiles in PCRF Core.

The **Diameter QoS** page allows you to create new and manage existing Diameter QoSs. The page displays the list of defined configurations and provides the options to import, export, or add data.

To confifure Diameter QoS:

1. From the navigation menu under **Traffic Profile**, select **Diameter QoS**.
   This opens the **Diameter QoS** page. The page lists the existing Diameter QoSs. You can add or import new data using this page.

> **Note**
>
> Click Export to download the available listings in the JSON file format on your system.

2. Click Add .

This opens the **Create Diameter QoS** page.

3. On the **Create Diameter QoS** page, enter values for the available input fields. The following table describes the fields:

Table 7-139 Create Diameter QoS Configurations

Field Name	Description
Name	Name of the Profile.
Description	Description of the profile.
QoS Class Identifier (1-254)	Identifies the QoS class. Enter a value between 1 and 254 or select from the following: • 1 = Conversational speech • 2 = Conversational • 3 = Streaming speech • 4 = Streaming • 5 = Interactive with priority 1 signalling • 6 = Interactive with priority 1 • 7 = Interactive with priority 2 • 8 = Interactive with priority 3 • 9 = Background • 65 = MC-PTT Voice • 66 = PTT Voice • 69 = MC-PTT Signaling • 70 = MC Data **Note:** QCI values between 1 and 254 are supported. Values other than 1 through 9, 65, 66, 69, and 70 are undefined.
Uplink Max Authorized Rate (bps)	Maximum authorized bandwidth in bits per second for uplinks (user equipment to network).
Downlink Max Authorized Rate (bps)	Maximum authorized bandwidth in bits per second for downlinks (network to user equipment).
Uplink Min Guranteed Rate (bps)	Minimum guaranteed bandwidth in bits per second for uplinks (user equipment to network). Only applicable if the QoS class identifier is between 1 and 4.
Downlink Min Guranteed Rate (bps)	Minimum guaranteed bandwidth in bits per second for downlinks (network to user equipment). Only applicable if the QoS class identifier is between 1 and 4.
ARP Priority Level	Allocation and Retention Priority level of the service flows associated with this profile. Highest: 1 Lowest: 15
ARP Preemption Capability	Enable or disable the ARP Preemption Capability. Select from the following: • PREEMPTION_CAPABILITY_ENABLED • PREEMPTION_CAPABILITY_DISABLED
ARP Preemption Vulnerability	Enable or disable the ARP Preemption Vulnerability. Select from the following: • PREEMPTION_VULNERABILITY_ENABLED • PREEMPTION_VULNERABILITY_DISABLED

Table 7-139 (Cont.) Create Diameter QoS Configurations

Field Name	Description
Resource Allocation Notification	Resource Allocation Notification Indicates that the allocation of resources for the related Diameter QoSs are confirmed. Select **ENABLE_NOTIFICATION** to enable.

4. Click **Save** to save the changes.

> **Note**
>
> Click **Cancel** to discard the changes.

The value gets listed on the **Diameter QoS** page. Use  or  available in the next column to update or delete the listing.

Importing Diameter QoS

To import Diameter QoS:

1. Click 

.
The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

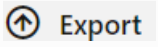
7.4.5.6.3 PCC Profile

This procedure provides information about how to define and manage PCC Profiles under Traffic Profiles in PCRF Core.

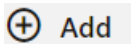
The **PCC Profile** page allows you to create new and manage existing PCC Profiles. The page displays the list of defined configurations and provides the options to import, export, or add data.

To confifure PCC Profile:

1. From the navigation menu under **Traffic Profile**, select **PCC Profile**.
This opens the **PCC Profile** page. The page lists the existing PCC Profiles. You can add or import new data using this page.

> **Note**
>
> Click 

to download the available listings in the JSON file format on your system.

2. Click 

.
This opens the **Create PCC Profile** page.
3. On the **Create PCC Profile** page, enter values for the available input fields.
The following table describes the fields:

Table 7-140 Create PCC Profile Configurations

Field Name	Description
Name	Name of the Profile.
Description	Description of the profile.
QoS Class Identifier (1-254)	Identifies the QoS class. Enter a value between 1 and 254 or select from the following: • 1 = Conversational speech • 2 = Conversational • 3 = Streaming speech • 4 = Streaming • 5 = Interactive with priority 1 signalling • 6 = Interactive with priority 1 • 7 = Interactive with priority 2 • 8 = Interactive with priority 3 • 9 = Background • 65 = MC-PTT Voice • 66 = PTT Voice • 69 = MC-PTT Signaling • 70 = MC Data **Note:** QCI values between 1 and 254 are supported. Values other than 1 through 9, 65, 66, 69, and 70 are undefined.
Bearer Usage	Indicates the Bearer usage. The default value is not applicable for this field. Users can select GENERAL or IMS_SIGNALLING from the drop down list as per the requirement.
Uplink Max Authorized Rate (bps)	Maximum authorized bandwidth in bits per second for uplinks (user equipment to network).
Downlink Max Authorized Rate (bps)	Maximum authorized bandwidth in bits per second for downlinks (network to user equipment).
Uplink Min Guranteed Rate (bps)	Minimum guaranteed bandwidth in bits per second for uplinks (user equipment to network).
Downlink Min Guranteed Rate (bps)	Minimum guaranteed bandwidth in bits per second for downlinks (network to user equipment).
ARP Priority Level	Allocation and Retention Priority level of the service flows associated with this profile. Highest: 1 Lowest: 15
ARP Preemption Capability	Enable or disable the ARP Preemption Capability. Select from the following: • PREEMPTION_CAPABILITY_ENABLED • PREEMPTION_CAPABILITY_DISABLED
ARP Preemption Vulnerability	Enable or disable the ARP Preemption Vulnerability. Select from the following: • PREEMPTION_VULNERABILITY_ENABLED • PREEMPTION_VULNERABILITY_DISABLED
Service Identifier	Credit-control service identifier associated with the traffic defined by this rule. Only applicable if online charging is enabled.

Table 7-140 (Cont.) Create PCC Profile Configurations

Field Name	Description
Rating Group	Credit-control rating group associated with the traffic defined by this profile. Only applicable if online charging is enabled.
Monitoring Key	Monitoring key that may apply to the PCC Rule.
Reporting Level	The reporting level. Select from the following: • SERVICE_IDENTIFIER_LEVEL • RATING_GROUP_LEVEL • SPONSORED_CONNECTIVITY_LEVEL
Online Charging	Specifies whether or not online charging is enabled in this profile. Select from the following: • DISABLE_ONLINE • ENABLE_ONLINE
Offline Charging	Specifies whether or not offline charging is enabled in this profile. Select from the following:: • DISABLE_OFFLINE • ENABLE_OFFLINE
Metering Method	Defines how service data-flow traffic is metered for offline charging. Select from the following: • DURATION • VOLUME • DURATION_VOLUME • EVENT
Precedence	Precedence value of the profile. The lower the precedence, the higher the priority.
Flow Status	Indicates whether the application traffic is enabled or disabled in either the uplink or downlink direction. Select from the following: • ENABLED_UPLINK • ENABLED_DOWNLINK • ENABLED • DISABLED
Resource Allocation Notification	Resource Allocation Notification Indicates that the allocation of resources for the related PCC rules are confirmed. Select **ENABLE_NOTIFICATION** to enable.
Required Access Info	Select from the following: • USER_LOCATION — the subscriber's location • MS_TIME_ZONE — the mobile subscriber's time zone • USER_LOCATION and MS_TIME_ZONE — the (mobile) subscriber's location and time zone If this field is not set, the device uses the values sent in AF requests; otherwise, it uses the values set here.
TDF Application Identifier	Identifies the traffic that belongs to the application to which the rule applies.

Table 7-140 (Cont.) Create PCC Profile Configurations

Field Name	Description
TDF Redirect Support	Indicates whether the application traffic should be redirected to another controlled address. Select from the following: • REDIRECTION_DISABLED • REDIRECTION_ENABLED
TDF Redirect Address Type	Specifies the format for the redirect address. Select from the following: • IPv4 • IPv6 • URL • SIP_URI
TDF Redirect Server Address	The address of the TDF redirect server in the specified address type.
Mute Notification	Notification to disable application detection notifications from the TDF device. Select **MUTE_REQUIRED** from the drop-down list. By default, mute remains disabled.
Sponsor Identity	Name identifying a connectivity sponsor.
Application Service Provider Identity	Name identifying an application service provider.
Flow Descriptions(s)	IP flows associated with this profile. A comma-separated list of Diameter IP Filter rules following the format specified in RFC 3588 section 4.3. Used in the following cases: • An old traffic profile is imported, and the flow description is not an empty string. • An upgrade from an older version is in process and the existing traffic profile flow description is not an empty string.

4. Click **Save** to save the changes.

> **Note**
>
> Click **Cancel** to discard the changes.

The value gets listed on the **PCC Profile** page. Use  or  available in the next column to update or delete the listing.

Importing PCC Profile

To import PCC Profile:

1. Click 

.
The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.5.6.4 PCC Rule

This procedure provides information about how to define and manage PCC Rules under Traffic Profiles in PCRF Core.

The **PCC Rule** page allows you to create new and manage existing PCC Rules. The page displays the list of defined configurations and provides the options to import, export, or add data.

To confifure PCC Rule:

1. From the navigation menu under **Traffic Profile**, select **PCC Rule**.
   This opens the **PCC Rule** page. The page lists the existing PCC Rules. You can add or import new data using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .

   This opens the **Create PCC Rule** page.

3. On the **Create PCC Rule** page, enter values for the available input fields.
   The following table describes the fields:

Table 7-141 Create PCC Rule Configurations

Field Name	Description
Name	Name of the Profile.
Rule Name	Uniquely identifies the PCC Rule. Used to reference an PCC Rule in communication between the CNC Policy and a PCEF within one IP-CAN session. **Note**: The ***Name*** and the ***Rule Name*** fields must have the same values. It helps in proper installation of PCC rules.
Description	Description of the profile.

Table 7-141 (Cont.) Create PCC Rule Configurations

Field Name	Description
QoS Class Identifier (1-254)	Identifies the QoS class. Enter a value between 1 and 254 or select from the following: • 1 = Conversational speech • 2 = Conversational • 3 = Streaming speech • 4 = Streaming • 5 = Interactive with priority 1 signalling • 6 = Interactive with priority 1 • 7 = Interactive with priority 2 • 8 = Interactive with priority 3 • 9 = Background • 65 = MC-PTT Voice • 66 = PTT Voice • 69 = MC-PTT Signaling • 70 = MC Data **Note:** QCI values between 1 and 254 are supported. Values other than 1 through 9, 65, 66, 69, and 70 are undefined.
Bearer Usage	Indicates the Bearer usage. The default value is not applicable for this field. Users can select GENERAL or IMS_SIGNALLING from the drop down list as per the requirement.
Uplink Max Authorized Rate (bps)	Maximum authorized bandwidth in bits per second for uplinks (user equipment to network).
Downlink Max Authorized Rate (bps)	Maximum authorized bandwidth in bits per second for downlinks (network to user equipment).
Uplink Min Guranteed Rate (bps)	Minimum guaranteed bandwidth in bits per second for uplinks (user equipment to network). Only applicable if the QoS class identifier is between 1 and 4.
Downlink Min Guranteed Rate (bps)	Minimum guaranteed bandwidth in bits per second for downlinks (network to user equipment). Only applicable if the QoS class identifier is between 1 and 4.
ARP Priority Level	Allocation and Retention Priority level of the service flows associated with this profile. Highest: 1 Lowest: 15
ARP Preemption Capability	Enable or disable the ARP Preemption Capability. Select from the following: • PREEMPTION_CAPABILITY_ENABLED • PREEMPTION_CAPABILITY_DISABLED
ARP Preemption Vulnerability	Enable or disable the ARP Preemption Vulnerability. Select from the following: • PREEMPTION_VULNERABILITY_ENABLED • PREEMPTION_VULNERABILITY_DISABLED
Service Identifier	Credit-control service identifier associated with the traffic defined by this rule. Only applicable if online charging is enabled.

Table 7-141 (Cont.) Create PCC Rule Configurations

Field Name	Description
Rating Group	Credit-control rating group associated with the traffic defined by this profile. Only applicable if online charging is enabled.
Reporting Level	The reporting level. Select from the following: • SERVICE_IDENTIFIER_LEVEL • RATING_GROUP_LEVEL • SPONSORED_CONNECTIVITY_LEVEL
Online Charging	Specifies whether or not online charging is enabled in this profile. Select from the following: • DISABLE_ONLINE • ENABLE_ONLINE
Offline Charging	Specifies whether or not offline charging is enabled in this profile. Select from the following:: • DISABLE_OFFLINE • ENABLE_OFFLINE
Metering Method	Defines how service data-flow traffic is metered for offline charging. Select from the following: • DURATION • VOLUME • DURATION_VOLUME • EVENT
Precedence	Precedence value of the profile. The lower the precedence, the higher the priority.
Flow Status	Indicates whether the application traffic is enabled or disabled in either the uplink or downlink direction. Select from the following: • ENABLED_UPLINK • ENABLED_DOWNLINK • ENABLED • DISABLED
Resource Allocation Notification	Resource Allocation Notification Indicates that the allocation of resources for the related PCC rules are confirmed. Select **ENABLE_NOTIFICATION** to enable.
Required Access Info	Select from the following: • USER_LOCATION — the subscriber's location • MS_TIME_ZONE — the mobile subscriber's time zone • USER_LOCATION and MS_TIME_ZONE — the (mobile) subscriber's location and time zone If this field is not set, the device uses the values sent in AF requests; otherwise, it uses the values set here.
TDF Application Identifier	Identifies the traffic that belongs to the application to which the rule applies.
TDF Redirect Support	Indicates whether the application traffic should be redirected to another controlled address. Select from the following: • REDIRECTION_DISABLED • REDIRECTION_ENABLED

Table 7-141 (Cont.) Create PCC Rule Configurations

Field Name	Description
TDF Redirect Address Type	Specifies the format for the redirect address. Select from the following: • IPv4 • IPv6 • URL • SIP_URI
TDF Redirect Server Address	The address of the TDF redirect server in the specified address type.
Mute Notification	Notification to disable application detection notifications from the TDF device. Select **MUTE_REQUIRED** from the drop-down list. By default, mute remains disabled.
Sponsor Identity	Name identifying a connectivity sponsor.
Application Service Provider Identity	Name identifying an application service provider.
PS to CS Session Connectivity	Indicates that the service data flow carries video and allows for packet switch (PS) to circuit switch (CS) session continuity. Select **VIDEO_PS2CS_CONT_CANDIDATE** from the drop-down list.
Flow Descriptions(s)	IP flows associated with this profile. A comma-separated list of Diameter IP Filter rules following the format specified in RFC 3588 section 4.3. Used in the following cases: • An old traffic profile is imported, and the flow description is not an empty string. • An upgrade from an older version is in process and the existing traffic profile flow description is not an empty string.

4. Click **Save** to save the changes.

> **Note**
>
> Click **Cancel** to discard the changes.

The value gets listed on the **PCC Rule** page. Use or available in the next column to update or delete the listing.

Importing PCC Rule

To import PCC Rule:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.5.6.5 Predefined ADC Rule

This procedure provides information about how to define and manage Predefined ADC Rules under Traffic Profiles in PCRF Core.

The **Predefined ADC Rule** page allows you to create new and manage existing Predefined ADC Rules. The page displays the list of defined configurations and provides the options to import, export, or add data.

To confifure Predefined ADC Rule:

1. From the navigation menu under **Traffic Profile**, select **Predefined ADC Rule**.
   This opens the **Predefined ADC Rule** page. The page lists the existing Predefined ADC Rules. You can add or import new data using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .

   This opens the **Create Predefined ADC Rule** page.

3. On the **Create Predefined ADC Rule** page, enter values for the available input fields. The following table describes the fields:

   Table 7-142 Create Predefined ADC Rule Configurations

Field Name	Description
Name	Name of the Predefined ADC Rule.
Rule-Base Name	Uniquely identifies the Predefined ADC Rule. Used to reference an Predefined ADC Rule in communication between the CNC Policy and a PCEF within one IP-CAN session.
Description	Description of the ADC Rule.

4. Click **Save** to save the changes.

   > **Note**
   >
   > Click **Cancel** to discard the changes.

   The value gets listed on the **Predefined ADC Rule** page. Use or available in the next column to update or delete the listing.

Importing Predefined ADC Rule

To import Predefined ADC Rule:

1. Click Import .
   The page opens the **File Upload** dialog box.

2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.5.6.6 Predefined ADC Rule Base

This procedure provides information about how to define and manage Predefined ADC Rule Bases under Traffic Profiles in PCRF Core.

The **Predefined ADC Rule Base** page allows you to create new and manage existing Predefined ADC Rule Bases. The page displays the list of defined configurations and provides the options to import, export, or add data.

To confifure Predefined ADC Rule Base:

1. From the navigation menu under **Traffic Profile**, select **Predefined ADC Rule Base**. This opens the **Predefined ADC Rule Base** page. The page lists the existing Predefined ADC Rule Bases. You can add or import new data using this page.

> **Note**
>
> Click Export to download the available listings in the JSON file format on your system.

2. Click Add .

   This opens the **Create Predefined ADC Rule Base** page.
3. On the **Create Predefined ADC Rule Base** page, enter values for the available input fields.
   The following table describes the fields:

Table 7-143 Create Predefined ADC Rule Base Configurations

Field Name	Description
Name	Name of the Predefined ADC Rule Base.
Rule-Base Name:	Uniquely identifies the Predefined ADC Rule Base. Used to reference an Predefined ADC Rule Base in communication between the CNC Policy and a PCEF within one IP-CAN session.
Description	Description of the Predefined ADC Rule Base.

4. Click **Save** to save the changes.

> **Note**
>
> Click **Cancel** to discard the changes.

The value gets listed on the **Predefined ADC Rule Base** page. Use or available in the next column to update or delete the listing.

Importing Predefined ADC Rule Base

To import Predefined ADC Rule Base:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.5.6.7 Predefined PCC Rule

This procedure provides information about how to define and manage Predefined PCC Rules under Traffic Profiles in PCRF Core.

The **Predefined PCC Rule** page allows you to create new and manage existing Predefined PCC Rules. The page displays the list of defined configurations and provides the options to import, export, or add data.

To confifure Predefined PCC Rule:

1. From the navigation menu under **Traffic Profile**, select **Predefined PCC Rule**.
   This opens the **Predefined PCC Rule** page. The page lists the existing Predefined PCC Rules. You can add or import new data using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .

   This opens the **Create Predefined PCC Rule** page.
3. On the **Create Predefined PCC Rule** page, enter values for the available input fields. The following table describes the fields:

Table 7-144 Create Predefined PCC Rule Configurations

Field Name	Description
Name	Name of the Predefined PCC Rule.
Rule Name	Uniquely identifies the Predefined Predefined PCC Rule. Used to reference an Predefined Predefined PCC Rule in communication between the CNC Policy and a PCEF within one IP-CAN session. **Note**: The ***Name*** and the ***Rule Name*** fields must have the same values. It helps in proper installation of PCC rules.
Description	Description of the Predefined PCC Rule.

4. Click **Save** to save the changes.

> **Note**
>
> Click **Cancel** to discard the changes.

The value gets listed on the **Predefined PCC Rule** page. Use  or  available in the next column to update or delete the listing.

Importing Predefined PCC Rule

To import Predefined PCC Rule:

1. Click  .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.5.6.8 Predefined PCC Rule Base

This procedure provides information about how to define and manage Predefined PCC Rule Bases under Traffic Profiles in PCRF Core.

The **Predefined PCC Rule Base** page allows you to create new and manage existing Predefined PCC Rule Bases. The page displays the list of defined configurations and provides the options to import, export, or add data.

To confifure Predefined PCC Rule Base:

1. From the navigation menu under **Traffic Profile**, select **Predefined PCC Rule Base**.
   This opens the **Predefined PCC Rule Base** page. The page lists the existing Predefined PCC Rule Bases. You can add or import new data using this page.

   > **Note**
   >
   > Click  to download the available listings in the JSON file format on your system.

2. Click  .
   This opens the **Create Predefined PCC Rule Base** page.
3. On the **Create Predefined PCC Rule Base** page, enter values for the available input fields.
   The following table describes the fields:

Table 7-145 Create Predefined PCC Rule Base Configurations

Field Name	Description
Name	Name of the Predefined PCC Rule Base.

Table 7-145 (Cont.) Create Predefined PCC Rule Base Configurations

Field Name	Description
Rule Name	Uniquely identifies the Predefined PCC Rule Base. Used to reference an Predefined PCC Rule Base in communication between the CNC Policy and a PCEF within one IP-CAN session. **Note**: The ***Name*** and the ***Rule Name*** fields must have the same values. It helps in proper installation of PCC rules.
Description	Description of the Predefined PCC Rule Base.
Monitoring Key	Monitoring key that may apply to the Predefined PCC Rule Base.

4. Click **Save** to save the changes.

> **Note**
>
> Click **Cancel** to discard the changes.

The value gets listed on the **Predefined PCC Rule Base** page. Use or available in the next column to update or delete the listing.

Importing Predefined PCC Rule Base

To import Predefined PCC Rule Base:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.4.6 Usage Monitoring

This section includes the configurations for Usage Monitoring.

To access Usage Monitoring configurations from the CNC Console home page, expand **Policy**, navigate to **Policy Data Configurations**, and select **Usage Monitoring**.

The data configuration for Usage Monitoring includes the following:

- Data Limit Profiles
- Data Limit Selection Profiles
- Data Limit Sorting Profiles
- Data Rollover Profiles

7.4.6.1 Data Limit Profiles

To configure the data limit profiles:

1. Navigate to **Data Limit Profiles** page under **Usage Monitoring** in **Policy Data Configurations**.

2. Click  .
The **Create Data Limit Profile** page appears.
3. On **Create Data Limit Profile** page, add the following details:

Field Name	Description
Profile Name	Uniquely identifies the UM Data Limit Profile by a name. This field is for configuration purpose only and not transmitted over the network.
Limit Identifier	Specifies the limit identifier. Multiple limit identifiers may have the same priority.
Monitoring Key	Uniquely identifies the UM Data Limit Profile by a Monitoring Key. If its value is not explicitly provided, then the data limit profile's limitId will be used as the monitoring Key of that profile.
Profile Type	Specifies the type of the Data Limit Profile. **Default value**: SUBSCRIBER **Note**: Striked-out entries are for future.
Plan Type	Specifies the type of Plan such as Base, Top-up, and Pass. This helps to determine the priority in selecting the plan over other types. The priority is configurable in service configuration screen. **Default value**: Base
Parent Plan	Identifies a Data Limit Profile as a parent data plan. Any attribute that is not explicitly configured in this plan shall be taken from the parent plan (if present).
Priority	A Priority is used to weigh it with other Data Limit Profiles and UDR provisioned Data Limits of the same Plan Type. The priority can be used by Policy or Data Limit Selection Profile for choosing among different candidate profiles. The Plan Type, Priority within that Plan Type and the Selection Order of Plan Types together decide the order of selection of Data Plans.
Validity Duration	Indicates the length of usage time in seconds. **Range**: 3600 - 31536000 (1 Year)
UM Level	Indicates the level of the usage monitoring instance (PDU Session level or per Service). **Default value**: Session Level
Inactivity Time (seconds)	Time interval in seconds after which the PGW or SMF shall stop time measurement for the Monitoring Key, if no packets are received belonging to the corresponding Monitoring Key **Range**: 0 - 604800 (7 Days) **Default value**: 86400 (1 Day)
Data Rollover Profile	A reference to a Data Rollover Profile. If not configured, Data Rollover will be disabled for this Data Plan.

Field Name	Description
Allow Excess Usage	Indicates whether or not excess usage is allowed for this Data Limit Profile. When Excess Usage is allowed, Usage Monitoring shall continue to grant and accumulate usage even after consuming 100% of allocated volume/time. The QoS, Charging etc. parameters can be controlled at PCRF Core Policy on account of the consumed usage. **Default value**: False
Usage Limit	
Duration	Indicates the length of time in seconds. **Range**: 3600 - 31536000 (1 Year)
Volume Total	Specifies the total data octets for both downlink and uplink **Range**: 512 - 1073741824000 (1000 GB)
Volume Uplink	Specifies the uplink data octets. **Range**: 512 - 1073741824000 (1000 GB)
Volume Downlink	Specifies the downlink data octets. **Range**: 512 - 1073741824000 (1000 GB)
Reset Period	
Period	Indicates whether the periodicity is "YEARLY", "MONTHLY", "WEEKLY", "DAILY" or "HOURLY". **Default value**: Monthly
Maximum Number Of Periods	Indicates the maximum number of periods after which the usage monitoring instance does not apply. If omitted, there is no limit in the number of periods.
Billing Day	
Type	Indicates which day of the period to reset the usage limit. **Default value**: LAST_DAY **Note**: If this configuration is updated, the new configuration shall be applied in the next billing cycle.
Day	Indicates the the specific day of the period. Day count starts from 1. If the Day entered is beyond the last day of a period (for example 31 for the month of April), the last day shall be considered. Required when "SPECIFIC_DAY" is selected in the Type field. **Note**: If this configuration is updated, the new configuration shall be applied in the next billing cycle.
Time	Indicates the specific time of day in "HH:MM:SS" **Default value**: "00:00:00" **Note**: If this configuration is updated, the new configuration shall be applied in the next billing cycle.
Excess Usage Limit	

Field Name	Description
Percentage	Indicates the percentage of base volume/time up to which excess usage is allowed. **Range**: 100 - 1500 **Note**: If all the Excess Usage Limit fields are left empty, the grant shall continue indefinitely.
Duration	Indicates the length of time in seconds. This is inclusive of the base duration. That is, the value in this field indicates the total duration to be consumed including base + excess. **Range**: 3600 - 31536000 (1 Year) **Note**: If all the Excess Usage Limit fields are left empty, the grant continues indefinitely.
Volume Total	Specifies the total data octets for both downlink and uplink. This is inclusive of the base total volume. **Range**: 512 - 1073741824000 (1000 GB) **Note**: If all the Excess Usage Limit fields are left empty, the grant continues indefinitely.
Volume Uplink	Uplink data octets. This is inclusive of the base uplink volume. **Range**: 512 - 1073741824000 (1000 GB) **Note**: If all the Excess Usage Limit fields are left empty, the grant continues indefinitely.
Volume Downlink	Downlink data octets. This is inclusive of the base downlink volume. **Range**: 512 - 1073741824000 (1000 GB) **Note**: If all the Excess Usage Limit fields are left empty, the grant continues indefinitely.

4. Click **Save** to save the Data Limit Profile.

7.4.6.2 Data Limit Selection Profiles

This page allows you to select, create, and configure Data Limit Selection Profiles.

Data Limit Selection is a set of rules to select one or more Data Limits based on certain conditions.

To create and configure the Data Limit Selection Profiles:

1. Navigate to **Data Limit Selection Profiles** page under **Usage Monitoring** in **Policy Data Configurations**.
2. Click Add .
   The **Create Data Limit Selection Profile** page appears.
3. Enter the a unique name to identify the Data Limit Selection Profile.
4. Click Add to create new selection rules.
   The **Add Selection Rules** dialog box appears.
5. Enter values for the available input fields described in the following table:

Table 7-146 Selection Rules Configurations

Field Name	Description
Priority	Indicates the selection rule priority.
Type	Indicates the selection rules type.
Data Limit Profile Name	This field is displayed when Type is "Data Limit Profile". This field should match the name configured in the Data Limit Profile.
UDR Data Limit Name	This field is displayed when Type is "UDR Data Limit". This field should match the Limit Identifier or the name of the plan if configured in Custom Attribute mapping and present in the UM Data Limit.
Always Match	When this is true, no conditions shall be evaluated. This can be used as a CATCH ALL Selection Rule. **Default value**: false

6. Under **Conditions**, click ⊕ Add .
   The **Add Conditions** dialog box appears.

   Enter values for the available input fields described in the following table:

Table 7-147 Conditions Configurations

Field Name	Description
Parameter Type	Indicates the type of parameter on which the condition will be applied. • Forwarded Attribute - Am attribute forwarded by the core service, for e.g. Serving Gateway MCCMNC / IP Address, APN etc. • Policy Decision Tag - A Tag applied by the policy.
Attribute Name	This field is displayed when Parameter Type is "Forwarded Attribute".
Operator	Accepts the following values: • Equals • Not Equals • Less Than • Greater Than • Matches
Value	Displays the list of configured Match Lists when the Operator is "Matches".

   Click **Save** to save the conditions.

7. Click **Save** to save selection rules.

7.4.6.3 Data Limit Sorting Profiles

This page allows you to select, create, and configure Data Limit Sorting Profiles.

Data Limit Sorting Rule is a set of rules to sort Data Limits. If Sorting Rules are configured, Sorting rules are applied after applying Selection Rules and when the output of Selection Rules results in more than one Data Limit.

To configure Selection and/or Sorting Rules to select one Usage Monitoring Data Limit out of many data limits provided by UDR or out of those configured in Data Limit Profiles, perform the following configurations:

1. Navigate to **Data Limit Sorting Profiles** page under **Usage Monitoring** in **Policy Data Configurations**.
2. Click Add .
   The **Create Data Limit Sorting Profile** page appears.
3. On the **Create Data Limit Sorting Profile** page:
   a. Enter the name of the Data Limit Sorting Profile.
   b. Under **Sorting Rules**, click Add .
      This opens the **Add Sorting Rules** dialog box.
   c. In **Add Sorting Rules** dialog box, perform the following configurations:

      Table 7-148 Sorting Rules Configurations

Field Name	Description
Index	Specifies the index. The sorting rules are applied in the order of index.
Parameter Type	Specifies the type of the parameter.
Attribute Name	Specifies the name of the attribute
Order	Specifies the order.

   d. Click **Save** to save sorting rules.
4. Click **Save** on the **Create Data Limit Sorting Profile** page to save the Data Limit Sorting Profile.

7.4.6.4 Data Rollover Profiles

To create the data rollover profile:

1. Navigate to **Data Rollover Profiles** page under **Usage Monitoring** in **Policy Data Configurations**.
2. Click Add .
   This opens **Create Data Rollover Profile** page.
3. In the **Create Data Rollover Profile** page, add the following configurations:

Field Name	Description
Profile Name	A unique to identify the Data Rollover Profile.
Data Rollover Enabled	Indicates whether to enable or disable Data Rollover.
Rollover Data Consumption	Indicates when to consume the rollover data. Rollover data to be consumed before or after a base plan. **Note**: As of now, Policy does not support consumption of the rollover data after the parent plan.

Field Name	Description
Maximum Number of Rollovers	Indicates how many cycles the data plan can rollover. A value of 0 means there is no limit to the number of cycles the left over data can rollover. **Range**: 0 to 24 **Default value**: 0 (No limit)
Data Rollover Percentage	Indicates the percentage of left over data volume/time to rollover to the next cycle. If rollover data is enabled, then data rollover percentage needs to be provided between 1-100. Also, if the plan contains timeThreshold, totalVolume, uplink and downlink then it should be advisable to provided all the fields in maximum and minimum roll-over for accurate calculation while creating and allocating roll-over data for a plan. **Range**: 0 to 100 **Default value**: 0.
Data Limit Cap Enabled	Indicates whether to enable or disable data limit cap. Data Limit Cap Settings is used to configure the data limit cap, which is applied on untill how much of data it must allow rollover across the cycles (rollover). Data Limit Cap has two components: • Base Data Limit, which is fixed. • Rollover, can accumulate over time and reach a value where the capping is applicable. Rollover configuration should be part of Rollover Profile. With different profiles, different capping schemes can be applied. For example, bounded and unbounded rollovers, single period rollover vs. multi-period rollovers. Data Limit Cap checks if the dataLimitCap - baseLimit - additionalRollOVer (which is calculated by adding all the roll-over across the reset cycle happens) > 0, then it allows for rollover. Otherwise, it does not allow the rollover. The maximum cap is applied on the total of base plan volume/time and total accumulated rolled over volume/time.
Maximum Rollover Threshold (To configure the maximum amount of data volume/time that can roll over from one cycle to another)	
Base Data Percentage	Indicates the percentage of base plan data volume/time that is allowed as rollover data volume/time. **Range**: 0 to 100
Duration	Indicates the length of time in seconds. **Range**: 3600 - 2592000 (30 Days)
Volume Total	Specifies the total data octets for both downlink and uplink. **Range**: 512 - 107374182400 (100 GB)

Field Name	Description
Volume Uplink	Specifies the uplink data octets. **Range**: 512 - 107374182400 (100 GB)
Volume Downlink	Specifies the downlink data octets. **Range**: 512 - 107374182400 (100 GB)
Minimum Rollover Threshold	
Duration	Indicates the length of time in seconds. The remaining duration if applicable shall be rolled over to the next period if it is greater than the threshold value configured here. **Range**: 0 - 2592000 (30 Days)
Volume Total	Specifies the total data octets for both downlink and uplink. The remaining total volume if applicable shall be rolled over to the next period if it is greater than the threshold value configured here. **Range**: 0 - 107374182400 (100 GB)
Volume Uplink	Specifies the uplink data octets. The remaining uplink volume if applicable will be rolled over to the next period if it is greater than the threshold value configured here. **Range**: 0 - 107374182400 (100 GB)
Volume Downlink	Specifies the downlink data octets. The remaining downlink volume if applicable will be rolled over to the next period if it is greater than the threshold value configured here. **Range**: 0 - 107374182400 (100 GB)
Data Limit Cap Settings Data Limit Cap Settings is used to configure the data limit cap, which is applied on untill how much of data it must allow rollover across the cycles (rollover).	
Duration	Indicates the length of time (duration) in seconds for which the data cap must be imposed. **Range**: 3600 - 2592000 (30 Days)
Volume Total	Indicates the maximum limit on the total volume of data octets for both downlink and uplink. **Range**: 512 - 107374182400 (100 GB)
Volume Uplink	Indicates the maximum volume of data that must be allowed for the uplink data octets. **Range**: 512 - 107374182400 (100 GB)
Volume Downlink	Indicates the maximum volume of data that must bbe allowed for the downlink data octets. **Range**: 512 - 107374182400 (100 GB)

4. Click **Save** to save the Data Rollover Profile.

7.5 Policy Management

Policy offers a Policy Design editor based on the Blockly interface. You can create and manage a Policy project for each of the following Policy services, based on your deployment:

- Session Management and Policy Authorization
- Access and Mobility Management

- UE Management
- PCRF Core
- Policy Data Source
- Usage Monitoring

This section describes how to create, manage, and deploy policies using the **Policy Management** pages.

The Policy Management configurations include:

- Policy Projects
- Policy Library
- Policy Tests

The following section describes the configurations on the **Policy Projects** page.

For information on **Policy Library** and **Policy Tests** functionalities, see *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.

7.5.1 Policy Projects

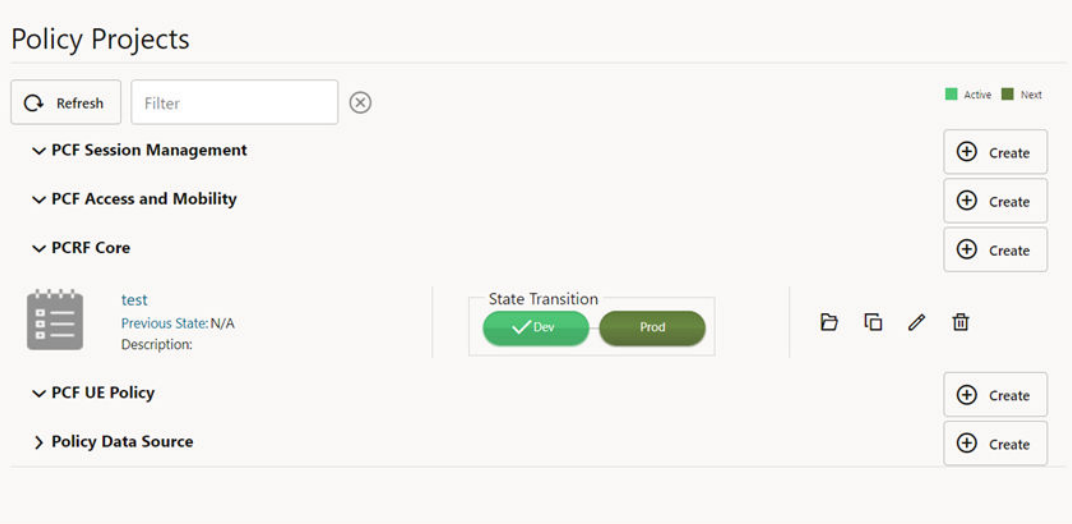
You can create and deploy a Policy project using **Policy Projects** page. There are two possible states for the Policy project, **Prod** and **Dev**.

To create and deploy a Policy project:

1. From the navigation pane under **Policy**, navigate to **Policy Management** and select **Policy Projects**.
   This opens the **Policy Projects** page, displaying the existing Policy projects for the respective Policy service.

   The following screen capture shows an illustration of the **Policy Projects** page:

Figure 7-28 Policy Projects



You can filter the existing projects based on Project Name and Active State using the Filter box available at the top of the page.

2. To create new project, click Create for the service group, for which the Policy project is to be created.
The page opens the **Create Policy Project** dialog box.

> **Note**
>
> Policy supports 10 projects per service type. An error message is displayed on clicking Create, if the number of projects created for a selected service type has reached maximum limit.

3. In the **Create Policy Projects** dialog box, enter the following information:

Field Name	Description
Name	The unique name you assign to the policy project This is a mandatory field. The name can only contain the characters A–Z, a–z, 0–9, period (.), hyphen (-), and underscore (_). The maximum length is 32 characters.
Description	Free-form text that identifies the Policy project. The maximum length is 255 characters.

4. Click **Save**.
This creates the Policy Project. The project is listed under the respective service name group.

> **Note**
>
> - Policy allows you to perform unit test for a project. For more information on testing the projects, see `Test Policy Projects` section in the *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.
> - The **Policy Project** page allows you to clone projects. Use the icon available with the Policy project listing to clone the project. The cloned project contains the data similar to the project from which it is cloned.
> - Use or to update or delete the Policy project.

5. To configure the Policy project, click corresponding to the project name.
This opens the Blockly editor. The editor allows you to construct one or more policies using the building blocks provided in the left hand side panel of the editor.

> **Note**
>
> The projects in **Prod** state are not editable. You can only view these projects and the associated policies, but cannot modify them.

For more information on configuring Policy blocks using Blockly editor, see *Oracle Communications Cloud Native Core, Converged Policy Design Guide*.

6. Change the state of a the Policy project, if required.
   Policy Projects can have any of the following states:
   - **Dev**: By default, the **Dev** state is assigned to a Policy project. Dev projects do not process any traffic in PRE.
   - **Prod**: The projects in **Prod** state processes traffic in PRE.

   The current state of any project can be identified with a tick mark and light green colored button as displayed at the upper right hand corner of the **Policy Projects** page.

   The following screen capture illustrates the different states of a project.

Figure 7-29 Policy Project States

> **Note**
>
> At any point of time there can be only one project in **Prod** state for a service. If you change the state of the project to **Prod** and there is already a project with the **Prod** state for that service, the **Prod** state for the existing project automatically gets updated to **Dev** state and the project in **Dev** state is updated to **Prod** state.

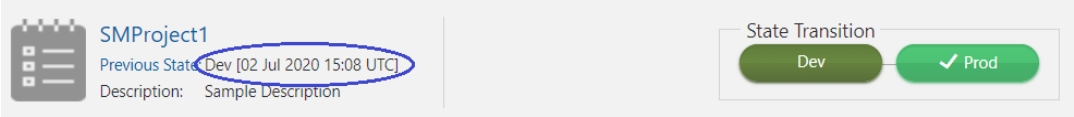
> **Note**
>
> When a user creates or updates (for example project states) a Policy project through CNC Console or REST API, it may take up to 10 seconds for the changes to reflect in the Policy rule.

Viewing Policy Project States

The **Policy Projects** page displays the previous states of projects along with the timestamp. The details are described in the following examples:

- **Example 1:**
  The following screen capture shows an example, displaying a project in **Prod** state:

Figure 7-30 Example 1



The following details are highlighted in the above example:

Current State: Prod

Previous State: Dev

Timestamp when the state changed from **Dev** to **Prod**: 02 Jul 2020 15:08 UTC

- **Example 2:**
  The following screen capture shows an example, displaying a newly created project in **Dev** state:

Figure 7-31 Example 2

The following points lists the details highlighted in the above example:

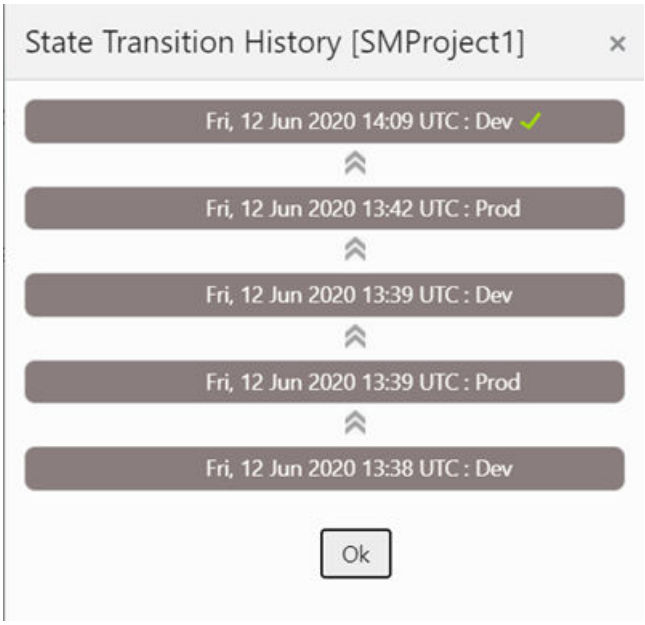
Current State: Dev

Previous State: N/A

Since it is a new project, there is no previous state.

- **Example 3:**
  The following screen capture shows an example of getting the complete history of the previous states of a Policy project by clicking the **Previous States**:

Figure 7-32 Example 3



7.6 Diameter Configurations

This section describes how to manage and view the Diameter Configurations in Policy using the **Diameter Configurations** pages.

7.6.1 Settings

The **Settings** page displays the general configurations related to the Diameter node. The page allows you to edit the configurations.

To edit settings:

1. From the navigation menu, under **Policy**, click **Diameter Configurations** and select **Settings**.
   This opens the **Settings** page. The page displays the existing configurations.
2. Click 

   .
   This opens the **Edit Settings** page.
3. Enter the following information under the respective groups:

Table 7-149 Edit Settings

Field Name	Description
Timer	
Reconnect Delay (sec)	Enter the time frame to delay before attempting to reconnect after a connection failure in seconds. The default value is 3 seconds.
Response Timeout (ms)	Enter the response timeout interval in milliseconds. The default value is 5000 ms. **Note**: The response timeout interval can have a decimal value. It helps to put the value in milliseconds.
Connection Timeout (sec)	Enter the connection timeout interval in seconds. The default value is 3 seconds.
WatchDog Interval (sec)	Enter the watchdog interval in seconds. The default value is 6 seconds.
Transport	
Protocol	The protocol supported is TCP.
Congestion Control	
Load Shedding Profile	Select any one of the configured load shedding profiles from the drop-down menu.
Message Priority Profile	Select any of the configured message priority profiles from the drop-down menu.
Overload Control	
Load Shedding Profile	Select any one of the configured load shedding profiles from the drop-down menu.

Table 7-149 (Cont.) Edit Settings

Field Name	Description
Message Priority Profile	Select any of the configured message priority profiles from the drop-down menu. **Note**: The following message priority data that was exported prior to Policy 23.2.0 cannot be imported as the data may be corrupt: • message containing Sd as interface • Sy-SLR as condition message The data with Sd interface or Sy-SLR condition messages that are exported only with Policy 23.2.0 or later versions can be imported.
Topology Hiding	
Topology Hiding	Enable or disable the topology hiding feature using this switch. When this feature is disabled, no changes are made to the orgin-host in the message. On enabling this feature, origin-host is updated in all or specific messages.
Apps to Hide	Specifies the application names for which topology hiding feature needs to be enabled. Users can select one or many values from the following list using the drop-down: • Rx • Gx • Sy • All When Topology Hiding feature is enabled with value `All` for **Apps to Hide** field, origin-host is replaced for all the diameter interface outgoing messages.
Enhanced Timer Configuration	
Application Name	Request Timer configuration for applications name like Rx, Gx, Sy, Sd.
Application Response Timeout (milliseconds)	Enter the application response timeout in milliseconds. The range of this value is between between 3 seconds to 2147483647.
Command Code Response Timeout	
AAR (milliseconds)	The command code response timeout value for AAR. The allowed value ranges from 3 to 2147483647. **Default Value**: 5000
STR (milliseconds)	The command code response timeout value for STR. The allowed value ranges from 3 to 2147483647. **Default Value**: 5000
RAR (milliseconds)	The command code response timeout value for RAR. The allowed value ranges from 3 to 2147483647. **Default Value**: 5000

Table 7-149 (Cont.) Edit Settings

Field Name	Description
ASR (milliseconds)	The command code response timeout value for ASR. The allowed value ranges from 3 to 2147483647. **Default Value**: 5000

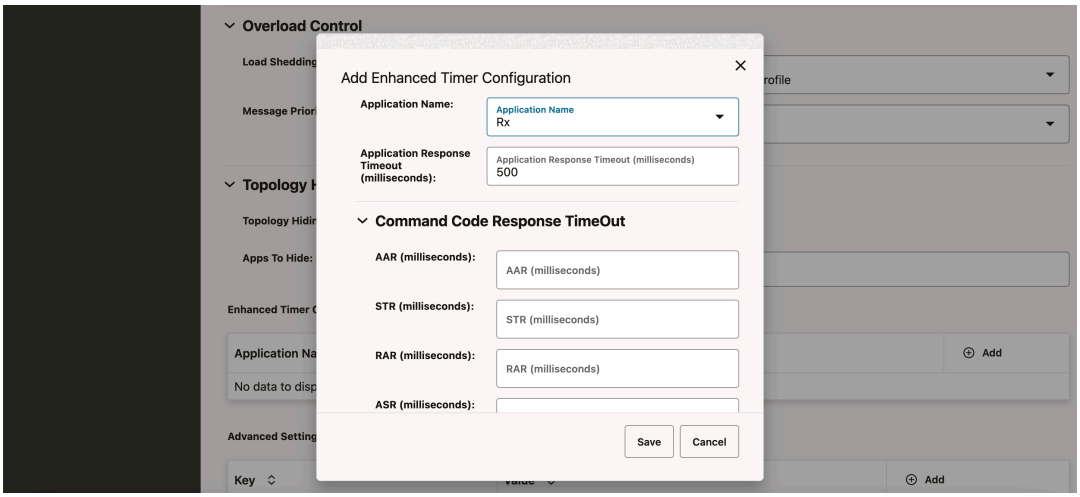
The order of precedence (from highest to lowest) of response timeout configurations is:

a. Command Code Response Timeout (ms) - Message level configurations i.e at AAR, STR etc.

b. Application Response Timeout (ms) - Interface level configurattion i.e at Gx, Rx etc.

c. Response Timeout (ms) - General level configutation

Figure 7-33 Add Enhanced Timer Configuration in the edit mode



4. Click **Save** to save the settings.

5. Perform the following steps to configure **Advanced Settings**:

a. Click the ⊕ Add
The page opens the **Add Advanced Settings** dialog box.

b. In the dialog box, enter the following **key** and respective **value**:

Table 7-150 Parameters for Advanced Settings

Keys	Value
DIAMETER.ErrorHandler.MaxRetryCount.Rx.RAR	It is used to set the maximum retries that can be performed for failed RAR messages. The retry attempt value can be any positive integer number between 1 to 2147483647. **Note**: If retry attempt value configured is a zero or negative number, then the default retry attempt value shall be considered. **Default Value**: 1

Table 7-150 (Cont.) Parameters for Advanced Settings

Keys	Value
DIAMETER.ErrorHandler.CycleBackRetry.Rx.RAR	It is used to set if peers can be cycled back for retries or not. **Default Value**: false
DIAMETER.ErrorHandler.MaxRetryCount.Rx.ASR	It is used to set the maximum retries that can be performed for failed ASR messages. The retry attempt value can be any positive integer number between 1 to 2147483647. **Note**: If retry attempt value configured is a zero or negative number, then the default retry attempt value shall be considered. **Default Value**: 1
DIAMETER.ErrorHandler.CycleBackRetry.Rx.ASR	It is used to set if peers can be cycled back for retries or not. **Default Value**: false
DIAMETER.gateway.binding.lookup.failure.forcerouting.enabled	This is used to enable force routing to PCRF Core and/or Diameter Connector on receiving Rx Commands along with Binding error response. **Default Value**: true
DIAMETER.gateway.binding.lookup.failure.forcerouting.service.list	This is used to set the force routing to PCRF Core and/or Diameter Connector. The possible list of values: • PCRF_CORE • DIAMETER_CONNECTOR
DIAMETER.gateway.binding.lookup.failure.forcerouting.exceptions	This is used to identify the exceptions for which force routing needs to be set. The possible list of values: • java.net.UnknownHostException • java.net.ConnectException • java.util.concurrent.TimeoutException But any **class exception** can be placed as a value of the list.
DIAMETER.gateway.binding.lookup.failure.forcerouting.responsecodes	This is used to identify the response error codes for which force routing needs to be set. The possible list of values: • 404 • 503 But any **http status code** can be placed as a value of the list.

c. Click **Save**.
The page saves the Error Handling configurations.

7.6.2 Peer Nodes

This procedure provides information about how to define and manage Peer Nodes in Diameter Configurations.

The **Peer Nodes** page allows you to create new and manage existing Peer Nodes. The page displays the list of defined configurations and provides the options to import, export, or add data.

To configure Peer Nodes:

1. From the navigation menu, under **Policy**, click **Diameter Configurations**, and select **Peer Nodes**.
   This opens the **Peer Nodes** page. The page lists the existing Peer Nodes. You can add or import new nodes using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .

   This opens the **Create Peer Node** page.

3. On the **Create Peer Node** page, enter values for the available input fields.
   The following table describes the fields:

Table 7-151 Create Peer Node Configurations

Field Name	Description
Name	Unique name of the peer node. Example value: ocs
Type	Defines which type of diameter service must be selected. The values can be • PCF • Application function (AF) • Backend • Diameter Routing Agent (DRA) • Online Charging System (OCS) • TDF • UDR You can choose a specific type if a particular NF can be directly connected. If NF is behind the DRA, then DRA must be chosen. Backend is a special type used by the engineering team to connect to the backend core services.
Reconnect Limit (sec)	The reconnect limit. This value must be configured as the Diameter peer configuration. Currently, all the Gateway pods reconnect for infinite time for the cloud native environment. Hence, this variable is of no use. This value can be kept as it is.

Table 7-151 (Cont.) Create Peer Node Configurations

Field Name	Description
Initiate Connection	Initiate connection has two options: • True: diameter-gateway initiates a connection. **Note:** Each diameter-gateway pod initiates one connection. • False: diamteter-gateway acts as a responder. It waits for other peers to initiate a connection. **Note:** When the diameter gateway is the connection initiator, then each diameter gateway pod would initiate 1 diameter connection with a network peer. For an example, if there are three diameter gateway pods and two external peers, then a mesh would be created with 6 diameter connections.
Port	Enter the port number. Enter a number from 0 to 65535. Example value: `8007`
Host	Enter the host name. Enter a FQDN or IP address available for establishing diameter transport connections to the peer node. **Note:** It is mandatory if *Initiate Connection* is set to true.
Realm	Enter the Realm of the peer to which diameter-gateway needs to be connected. For example,to add the realm detail of the OCS peer, enter `oracle.com` .
Identity	Enter an identity of the peer to which diameter-gateway needs to be connected. For example, to add the identity detail of the OCS peer, provide value enter `ocs`.

4. Click **Save** to save the changes.

> **Note**
>
> Click **Cancel** to discard the changes.

The value gets listed on the **Peer Node** page. Use or available in the next column to update or delete the listing.

Importing Peer Node

To import peer node:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.6.3 Routing Table

Configuration allows the routing of Diameter request messages to next hop peer based on `Diameter application-id`, `Destination-Realm`, and `Destination-Host` using Diameter routing table.

When using routing table, there are two ways to configure the next hop route:

1. Host-Based Routing: The destination-host of incoming message is checked in the routing table, and then the message is routed to the top priority of the matching route's peer.
2. Realm-Based Routing: The destination-realm of incoming message is checked in the routing table, and then the message is routed to the top priority of the matching route's peer.

Routing decision at Diameter-Gateway

Diameter gateway follows below steps in the following sequence:

1. If the incoming request message has destination-host and the specified peer is directly connected with the gateway pod, then the message is routed to the peer specified in destination-host.
2. If the incoming request message has destination-host and is not directly connected via any other diameter-gateway pods in cluster, then the message will be inter-pod routed.
3. The routing table is scanned for a matching route:
   - If the host is reachable, message is sent.
   - If the host is not reachable directly, find if it can be reached by another diameter gateway pod, message is sent using inter-pod route.
   - If the host is not reachable directly or indirectly, look up the routing table again for the next priority matching route.

The **Diameter Routing Table Configurations** page displays the Diameter routing table configurations. The page allows you to edit the configurations.

To configure the diameter routing table:

1. From the navigation menu, under **Policy**, click **Diameter Configurations** and select **Routing Table**.
   This opens the **Diameter Routing Table Configurations** page. The page displays the existing configurations.
2. Click Edit .
   This opens the **Edit Diameter Routing Table Configurations** page.
3. Expand the **Diameter Route Table** Table group.
   The expanded group allows you to add route table entries.
4. To add routing table:
   a. Click Add .
      The page opens the **Add Diameter Route Table** dialog box.
   b. Enter the values for the following input fields:

Table 7-152 Add Diameter Route Table Configuration

Field Name	Description
Priority	Defines the order of use when one or more routes have overlapping criteria. The range is 0-65535. Example value: `1`
Name	Configure Name as the value specified in Server Identifier. Starting with release 1.11.1, for the data sources configured as peer nodes in diameter gateway for Sy interface, the diameter routing table configurations must be added. While configuring, ensure that the given name is same as the value specified in the Server Identifier field.
Type	Route type. The value can be Realm or Host. • When the **Realm** is selected in the **Type** field, **Realms** field is displayed. • When the **Hosts** is selected in the **Type** field, **Hosts** field is displayed.
Realms	The realm detail of the route. Example value: `oracle.com`
Application ID	Select Rx , Gx, Sy, or All. Example value: `Sy`
Server Identifier	Specifies the server to which the message is to be routed. This identity must also be present in the Identity field of the peer node.

c. Click **Save** on the **Add Diameter Routing Table** dialog box.

5. On the **Edit Diameter Routing Table Configurations** page, expand the **Default Route** group.
6. Enter value for the **Server Identifier** drop-down list.
   The server identifier drop-down list shows the list of the configured peer nodes on the **Peer Nodes** configuration page. For more information on configuring Peer Nodes, see Peer Nodes.

   On selecting any of the value, ensure that the name is same as the values of server identifier.

> **Note**
>
> * (asterisk) wildcard character is allowed in **Hosts**, **Realms**, and **Server Identifier** fields.

7. Click **Save**.
   The configuration gets listed on the **Diameter Routing Table Configurations** page.

> **Note**
>
> Use  or  available under the **Actions** column to update or delete the Diameter Routing Table configurations.

7.6.4 Peer Node Sets

To configure Peer Node Sets:

1. From the navigation menu, under **Policy**, click **Diameter Configurations** and select **Peer Node Sets**.
   This opens the **Peer Node Sets** page. The page lists the existing Peer Node Sets. You can add or import new nodes using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .

   This opens the **Create Peer Node Sets** page.

3. On the **Create Peer Node Sets** page, enter values for the available input fields.
   The following table describes the fields:

Table 7-153 Create Peer Node Sets Configurations

Field Name	Description
Name	Unique name of the peer node. Example value: `ocs`
Description	
Type	Defines which type of diameter service must be selected. The values can be • PCF • Application function (AF) • backend • diameter routing agent (dra) • online charging system (ocs) • tdf • udr
Realm	Enter the realm name, that is, FQDNs to all of that computers that transact diameter traffic. For example,to add the realm detail of the OCS peer, enter `oracle.com` .
Primary Server	The primary server for a TDF client.
Secondary Server	The secondary server for a TDF client.
Tertiary Server	The teritiary server for a TDF client.
Quaternary Server	The quatenary server for a TDF client.

4. Click **Save** to save the changes.

   > **Note**
   >
   > Click **Cancel** to discard the changes.

The value gets listed on the **Peer Node Sets** page. Use or available in the next column to update or delete the listing.

7.6.5 Diameter Error Configurations

This section describes how to customize the Diameter error codes according to the network requirements using the Diameter Error Codes page.

The **Diameter Error Codes** page on CNC Console allows users to view and edit conditions defined by default for the Policy network function. This page also provides the options to import and export Diameter error codes.

The following table describes the errors supported by Policy in Rx application ID:

Table 7-154 Error Codes and Responses

Condition ID and Name	Error Message	Diameter Error Code/ Experimental Result Code	Application Error Code
Invalid AVP Value	The request contained an AVP with an invalid value in its data portion.	5004	DIAMETER_INVALID_AVP_VALUE
Missing AVP	The request did not contain an AVP that is required by the Command Code definition.	5005	DIAMETER_MISSING_AVP
Unsupported Command Code	The Request contained a Command-Code that the receiver did not recognize or support.	3001	DIAMETER_COMMAND_UNSUPPORTED
Loop Detected	The Request contained a Command-Code that the receiver did not recognize or support.	3005	DIAMETER_LOOP_DETECTED
Invalid Diameter Error	A request was received whose bits in the Diameter header were either set to an invalid combination, or to a value that is inconsistent with the command code definition. **Note:** Strict parsing must be enabled.	3008	DIAMETER_INVALID_HDR_BITS
Invalid AVP Bits	A request was received that included an AVP whose flag bits are set to an unrecognized value, or that is inconsistent with the AVP definition. **Note:** Strict parsing must be enabled.	3009	DIAMETER_INVALID_AVP_BITS

Table 7-154 (Cont.) Error Codes and Responses

Condition ID and Name	Error Message	Diameter Error Code/ Experimental Result Code	Application Error Code
Unsupported AVP	The peer received a message that contained an AVP that is not recognized or supported and was marked with the Mandatory bit. **Note:** Strict parsing must be enabled.	5001	DIAMETER_AVP_UNSUPPORTED
AVP Occurs Too Many Times	A message was received that included an AVP that appeared more often than permitted in the message definition. **Note:** Strict parsing must be enabled.	5009	DIAMETER_AVP_OCCURS_TOO_MANY_TIMES
Unsupported Version	A request was received, whose version number is unsupported.	5011	DIAMETER_INVALID_AVP_LENGTH
Invalid AVP Length	The request contained an AVP with an invalid length. **Note:** Strict parsing must be enabled.	5014	DIAMETER_INVALID_AVP_LENGTH
IP-CAN Session Not Available	A request is received without associated PDU session for a binding.	5065	IP-CAN_SESSION_NOT_AVAILABLE
Unknown Session ID	A request is received with unknown session ID.	5002	DIAMETER_UNKNOWN_SESSION_ID
Internal Error	An internal failure occurred.	5012	DIAMETER_UNABLE_TO_COMPLY

The following table describes the errors supported by Policy in CapEx application ID:

Table 7-155 Error Codes and Responses

Condition ID and Name	Error Message	Diameter Error Code	Application Error Code
No common security	CER message is received, and there are no common security mechanisms supported between the peers.	5017	DIAMETER_NO_COMMON_SECURITY
No common security	CER message is received, and there are no common applications supported between the peers.	5010	DIAMETER_NO_COMMON_APPLICATION

Table 7-155 (Cont.) Error Codes and Responses

Condition ID and Name	Error Message	Diameter Error Code	Application Error Code
No common security	Received the diameter request for unsupported application.	3007	ERROR_COND_APPLICATION_UNSUPPORTED

Editing Diameter Error Codes

To edit any of the defined conditions, perform the following steps:

1. From the navigation menu, click **Policy**, then select **Diameter Configurations**, and click **Diameter Error Codes**.
   This opens the Diameter Error Codes page that lists the **CapEx** and **Rx** application ID.
2. Click **Edit** against the application ID that you need to customize.
   This opens the Edit Diameter Error Codes page for the selected application ID.
3. Update the required values for the fields as described in the following table:

Table 7-156 Parameters for Edit Diameter Error Codes

Parameter	Description	Applicable for
Condition Name	Specifies the description for a defined condition. It is recommended to use descriptions that clearly explain the condition.	Rx and CapEx
Result Code	Specifes the Diameter result code for a defined condition. When Use Experimental Result switch is disabled, this field cannot be left blank. **Note:** The value must be a standard diameter result code as defined in the RFC 6733.	Rx and CapEx
Use Experimental Result	Indicates whether to use the Result Code AVP (268) or Experimental Result AVP (297) when an error result is generated by Policy.	
Vendor ID	Specifies the Vendor ID of the operator or governing body that manages the code entered by the user in the **Experimental Result Code** field. When **Use Experimental Result** switch is enabled, this field cannot be left blank.	Rx
Experimental Error Code	Specifies the custom Diameter result code for a defined condition. When Use Experimental Result switch is enabled, this field cannot be left blank. **Note:** The value must be a standard diameter result code, from 3000 to 9999, as defined in the 3GPP Technical Specification 29.230.	Rx
Error Message	A message that explains the nature of the error. This error message is only for user understanding and must not be parsed by network entities.	Rx and CapEx

   Use  available under the **Actions** column to update the error code.
4. Click **Save**.

Unsupported Diameter Error Codes

The following table describes the Diameter Error Codes that are not supported while configuring the Diameter Error codes feature:

Table 7-157 Unsupported Diameter Error Codes

Status Code	Description
3006	DIAMETER_REDIRECT_INDICATION
3010	DIAMETER_UNKNOWN_PEER
4001	DIAMETER_AUTHENTICATION_REJECTED
4002	DIAMETER_OUT_OF_SPACE
4003	ELECTION_LOST
5006	DIAMETER_RESOURCES_EXCEEDED
5007	DIAMETER_CONTRADICTING_AVPS
5013	DIAMETER_INVALID_BIT_IN_HEADER
5015	DIAMETER_INVALID_MESSAGE_LENGTH
5016	DIAMETER_INVALID_AVP_BIT_COMBO
5061	INVALID_SERVICE_INFORMATION
5062	FILTER_RESTRICTIONS
5063	REQUESTED_SERVICE_NOT_AUTHORIZED
5064	DUPLICATED_AF_SESSION
5066	UNAUTHORIZED_NON_EMERGENCY_SESSION
5067	UNAUTHORIZED_SPONSORED_DATA_CONNECTIVITY
5068	TEMPORARY_NETWORK_FAILURE
5198	DIAMETER_OVERLOAD_RETRY_NOT_ALLOWED_TO_ANY
5199	DIAMETER_NEWER_SESSION_DETECTED
5999	ALLOWED_SERVICE_NO_POLICIES_REQUIRED
5003	DIAMETER_AUTHORIZATION_REJECTED
3003	DIAMETER_REALM_NOT_SERVED

7.7 Data Source Configurations

Policy establishes connections with data sources to retrieve information about subscribers from the database. It queries a data source using a key attribute that uniquely identifies a subscriber and stores the results in the cache. A data source uses this key attribute, such as the phone or account number of the subscriber to index the information available in the database.

This section describes how to manage and view the Policy datasource configurations in Policy, using the **Data Source Configurations** pages.

7.7.1 Data Sources

This procedure provides information about how to configure and manage the Policy data sources.

The **Data Sources** page allows you to create new and manage existing data sources. The page displays the list of defined data source configurations and provides the options to import, export, or add configurations.

To configure data source:

1. From the navigation menu under **Policy**, navigate to **Data Source Configurations**, and select **Data Sources**.
   This opens the **Data Sources** page. The page lists the existing data source configurations. You can add or import new data sources using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .
   This opens the **Create Data Source** page.
3. On the **Create Data Source** page, enter the information specific to the available groups:

Table 7-158 Create Data Source Configuration

Field Name	Description
Name	Specifies the name of the data source. Example value: `ocs`
Description	Specifies additional information about the data source.
Type	Specifies the type of data source. Users can select any of the following valid values from the drop-down menu: • LDAP • Sy
Read Connection	On selecting LDAP as **Type**, this field becomes available. It is used to specify the number of read connections established with the data source.
admin state	Enable this switch for data source (Sy) interaction to happen. This switch becomes available when **Type** is Sy. When the admin state switch is disabled, the data source becomes unavailable temporarily.
Realm	Specifies the realm of the primary and optional secondary servers to connect. In case of Sy data sources, the `DestinationRealm AVP` in request message to Diameter Gateway is set using this value. This field becomes available when **Type** is Sy.
Select OCS from Peer Configurations	When this switch is enabled, the user can configure primary, secondary, tertiary and quaternary servers by selecting diameter peer nodes using the drop-down menu. For the peer nodes to be made available while configuring a data source, the user must have performed the configurations on Peer Nodes page under **Diameter Configurations**. This switch becomes available when **Type** is Sy.
Primary Server: The primary data source server details.	

Table 7-158 (Cont.) Create Data Source Configuration

Field Name	Description
Identity	Primary server identity. Example value: `oc-diam-gateway`
Addr	Load balancer IP of Diameter gateway.
Port	Primary server Port.
Secondary Server The secondary data source server details. If primary server is not reachable, then data source connection is established with secondary (if available). You can add or remove the secondary data source server.	
Identity	Secondary server identity. Example value: `oc-diam-gateway`
Addr	Load balancer IP of Diameter gateway.
Port	Secondary server Port.
Tertiary Server The tertiary data source server details. Tertiary data source server. If primary and secondary are not reachable, then LDAP connection is established with tertiary (if available). You can add or remove the tertiary data source server.	
Quaternary Server Quaternary data source server. If primary , secondary and tertiary are not reachable then LDAP connection is established with tertiary (if available). You can add or remove the quaternary data source server.	
Search Criteria	The criteria on which the data source search is performed. You can add or remove the Search Criteria settings. **Note:** For LDAP data source, enter **Root DN** and **Base DN Attribute**. **root dn** is the root of the LDAP tree in which to search and **Base DN Attribute** maps the key.
Search Filter	The search filter during the data source search. You can add or remove the **Search Fiter** settings.

4. Click **Save**.
   The Data source gets listed on the **Data Sources** page.

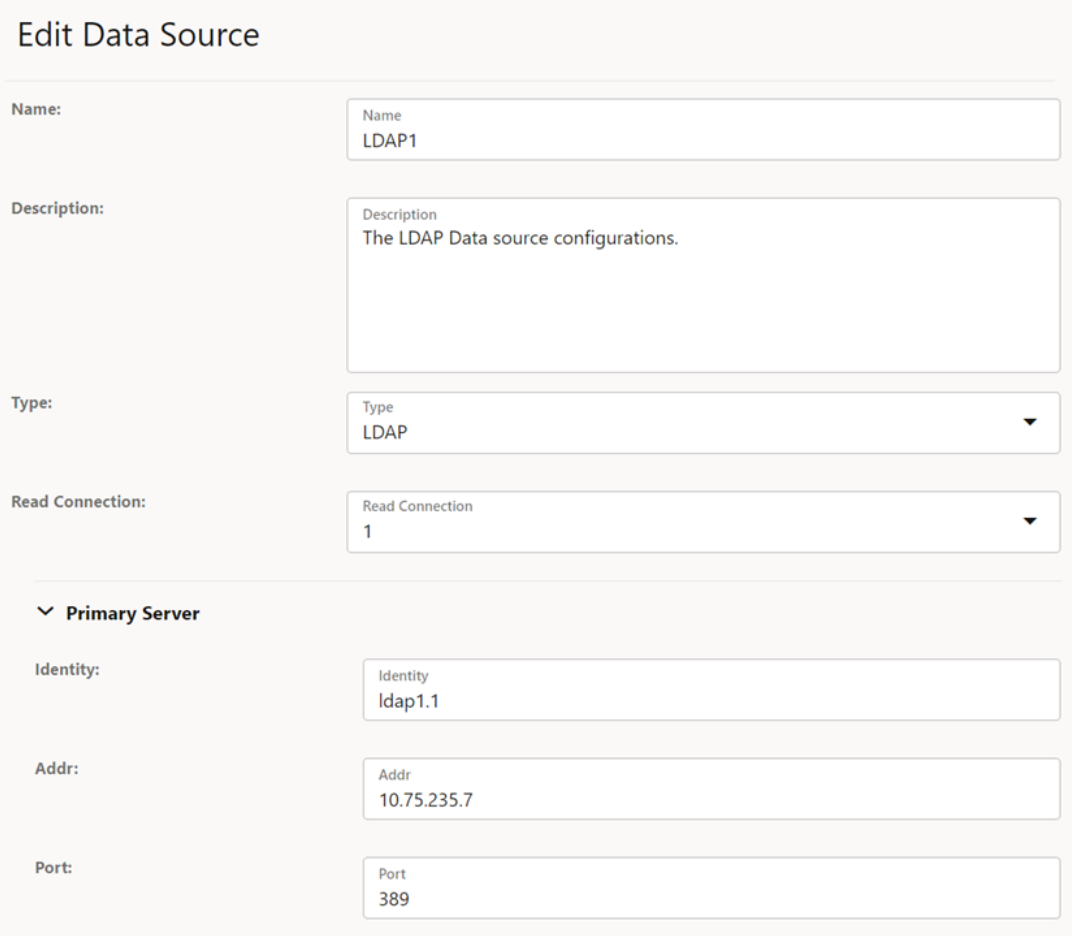
> **Note**
>
> Use  or  available under the **Actions** column to update or delete the data source configuration.

Setting Up LDAP as Policy Data Source

To set LDAP as the data source:

1. On the **Create Data Source** page, in the **Type** drop-down list, select LDAP.
2. Enter the **Primary Server** configurations.
   The following screen capture shows an example on how to create LDAP data source with name **LDAP1**:

Figure 7-34 LDAP Configuration using Edit Data Sources Page



3. Click **Save**.
   This creates the LDAP1 data source.
4. Create the **pds** service type in PCF system using the **Policy Engine** page.
   For more information about creating **pds** service type, see Policy Engine.

> **Note**
>
> On the **Policy Engine** page, the service name must be entered as **pds**.

5. Create a Policy Project for the **PDS** service type.
   For information about creating Policy Projects, see Adding PDS Service using Policy Engine.
6. Create policy action and condition in the newly created policy project for **PDS** service type.
   For detailed information about configuring the policy blocks for PDS service type, see *Oracle Communication Cloud Native Core, Converged Policy Design Guide*.

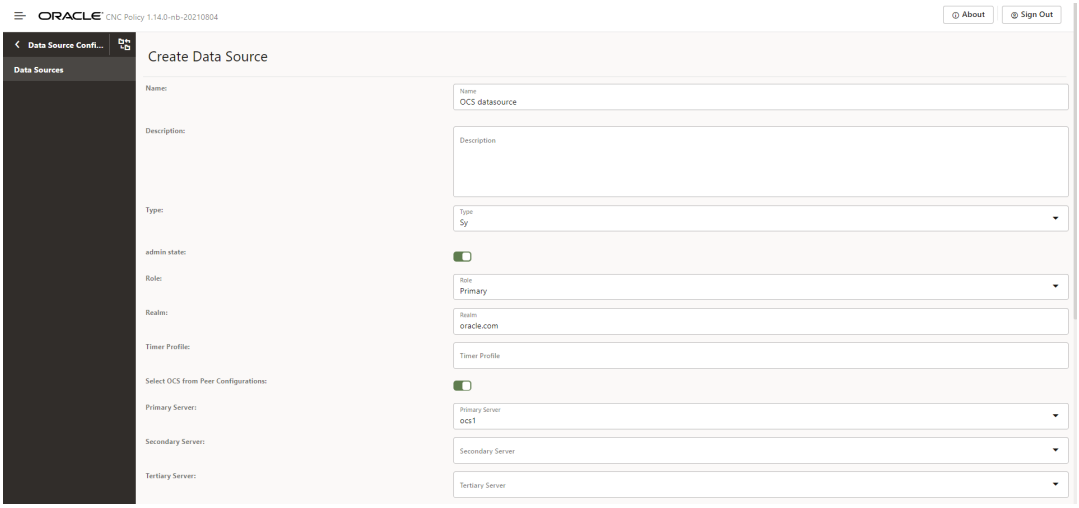
Setting Up OCS as Policy Data Source

To set OCS as the data source, perform the following steps:

1. On the **Create Data Source** page, in the **Type** drop-down list, select Sy.

2. Select the **Primary Server** using the drop-down menu. The peer nodes are configured using the Peer Nodes page under **Diameter Configurations**.
   The following screen capture shows an example on how to create OCS data source with name **OCS datasource**:

Figure 7-35 OCS Configuration using Edit Data Sources Page



3. Click **Save**.
   This creates the `OCS datasource` .
4. Create the **pds** service type in PCF system using the **Policy Engine** page.
   For more information about creating **pds** service type, see Policy Engine.

> **Note**
>
> On the **Policy Engine** page, the service name must be entered as **pds**.

5. Create a Policy Project for the **PDS** service type.
   For information about creating Policy Projects, see Adding PDS Service using Policy Engine.
6. Create policy action and condition in the newly created policy project for **PDS** service type.
   For detailed information about configuring the policy blocks for PDS service type, see *Oracle Communication Cloud Native Core, Converged Policy Design Guide*.

7.8 Administration

This section describes how to perform administration tasks, such as bulk import and bulk export of configurable objects into the Policy system.

7.8.1 Import & Export

This section describes how to perform the bulk export or bulk import of the managed objects (MOs) configured on Policy.

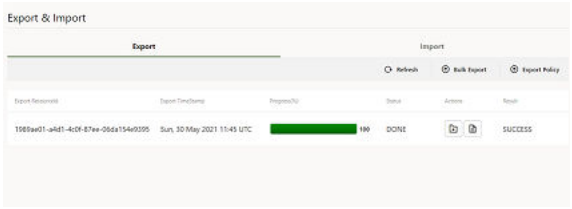
You can perform the bulk export and import of Policy data using the following methods:

- **Using CNC Console for Policy**:
  Policy provides the GUI to perform bulk export and import of Policy data.

To access the export and import functionality from the CNC Console home page, expand **Policy**, navigate to **Administration**, and select **Import & Export**.

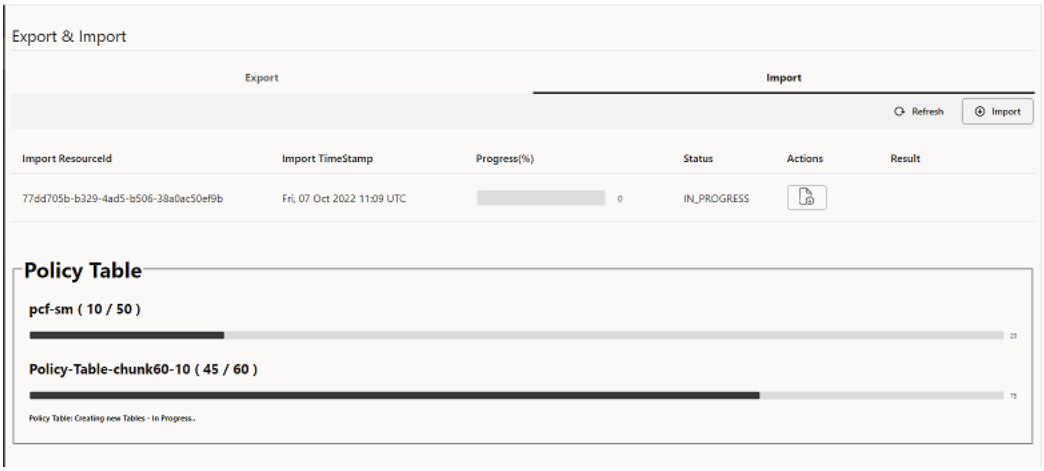
The page displays the **Export** and **Import** tabs. By default, the **Export** tab remains selected. The following screen capture illustrates an example of the **Import & Export** page:

Figure 7-36 Import and Export



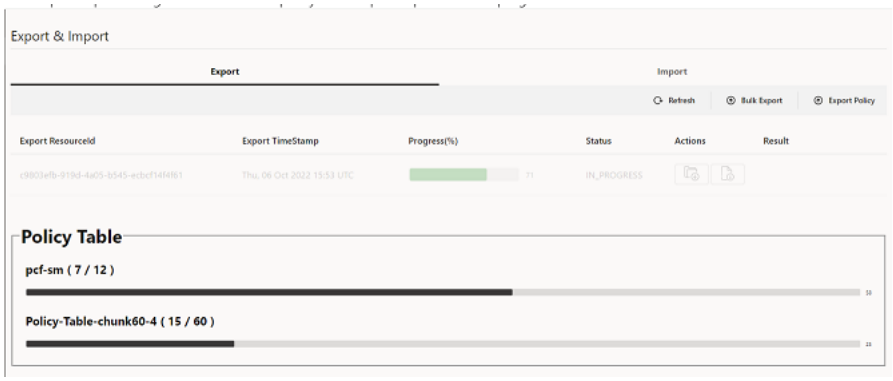
On initiating either bulk import or export for managed objects (MO's), a detailed progress information is displayed in the UI. Information like managed object type being imported or exported, total number of processes or tables, current number of managed object type being processed, managed object name like table name or rule name, number of records processed for managed Object, total number of managed object records present and progress messages. The following screen capture illustrates an example of the **Import & Export** page with detailed progress description:

Figure 7-37 Import showing Managed Object count progress status



> **Note**
>
> – Importing the Overload Threshold Profile will be rejected if the CPU validation mentioned under `Configure Threshold Values` section in Overload Control Threshold fails for any of the three threshold levels.
> – The service names mentioned in the json file used to import the Overload Threshold Profile must be same as mentioned in the exported json file.
> – The default Overload Threshold Profile cannot be exported or imported. The default Overload Threshold Profile must be customized with a different profile name before exporting.

Figure 7-38 Export showing Managed Object count progress status



> **Note**
>
> Currently this detailed progress information display is supported for export/import of records from Policy Table MO only.

You can perform the following operations using the **Import & Export** page:

– Exporting Policy Data
– Importing Policy Data

> **Note**
>
> The following message priority data that was exported prior to Policy 23.2.0 cannot be imported as the data may be corrupt:
>
> – message containing Sd as interface
> – Sy-SLR as condition message
>
> The data with Sd interface or Sy-SLR condition messages that are exported only with Policy 23.2.0 or later versions can be imported.

- **Using REST API for Policy**:
  Policy provides REST APIs to bulk export and import of Policy data. For more information about REST API configuration, see Using REST API for Policy Import & Export.

7.8.1.1 Exporting Policy Data

The export functionality allows you to export Policy configurations and Policy Projects with the respective dependencies. You can use the export functionality to perform the following operations:

- Exporting Policy Configurations with Dependencies
- Exporting Policy Projects with Dependencies

The Policy data export is aligned with the left navigation menu options under **Policy**, on the CNC Console. You can export either all the configurations or the configurations of the selected menu options.

To export the policy configuration data:

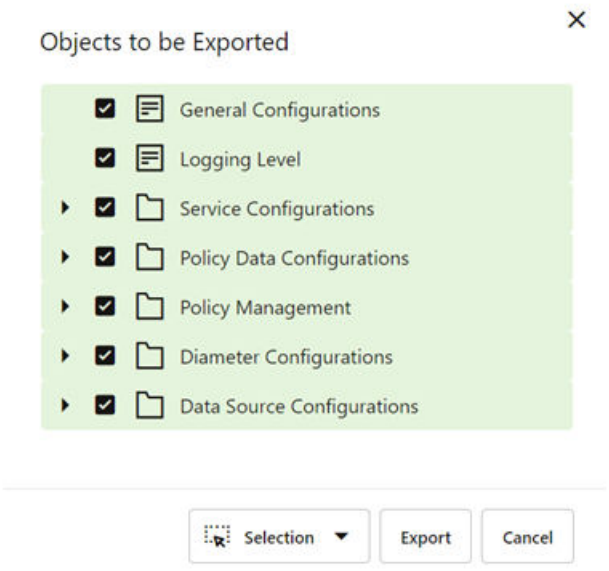
1. From **Policy**, navigate to **Administration** and select **Import & Export**.

   This opens the **Import & Export** page, displaying the **Export** and **Import** tabs. By default, the **Export** tab remains selected.

2. Click Bulk Export .

   This opens the **Objects to be Exported** dialog box, displaying the list of Policy configurations in a menu tree structure. The dialog box allows you to select the configurations to be exported. The following screen capture displays an illustration of the **Objects to be Exported** dialog box:

Figure 7-39 Objects to be Exported dialog box

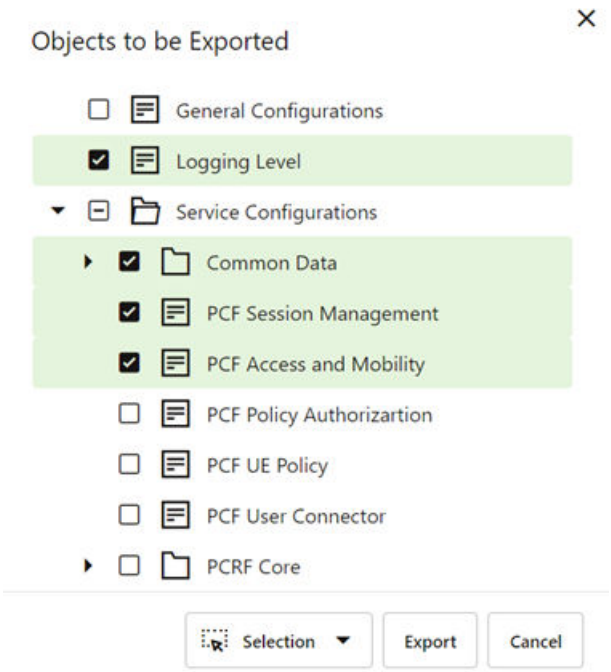


   By default, the **Objects to be Exported** dialog box has all the configurations selected for export. You can deselect the configurations.

3. In the dialog box, select or deselect the configurations according to the export requirement.

   Selecting or deselecting a parent folder, automatically selects or deselects all the child nodes respectively. In case a folder is selected partially, then a partial selection sign is displayed on the parent folder as shown in the following illustration:

Figure 7-40 Objects to be Exported dialog box with selected configurations



> **Note**
>
> To select or deselect all the configurations, click **Selection** and perform the required operation.

4. Click **Export**.

> **Note**
>
> Click **Cancel** to discard the export operation.

This starts the export of all the configurations related to the selected MOs. A row is created in the export status table on the **Import & Export** page, displaying the export status with the following details:

- **Export ResourceId**: A new export resource ID is generated for each export operation. You can use this ID to get the export status.
- **Creation TimeStamp**: The timestamp of generation of **Export ResourceId**.
- **Progress (%)**: Shows the export progress in form of a percentage bar. The page auto refreshes the status, until the progress reaches 100 percent.
- **Status**: The status of the export operation. The status can be any of the following:
  - **INIT**: The validation of policies is in progress and the export of the MOs is not yet started.
  - **IN_PROGRESS**: The export is running.

  - **DONE**: The export is complete.
- **Actions**: Provides the buttons to download the following:
  - **Export configuration files in ZIP file format**: The exported configurations in ZIP file format. The ZIP file contains the MO data in JSON file format. You can download the exported data by clicking  under the **Action** column.
  - **Export report in TEXT file format**: The export report provides results for each JSON file present in the exported ZIP file. It also provides the reason for failure, in case the export of any of the configuration fails.

    You can download the export report by clicking  under the **Action** column.

    > **Note**
    >
    > The buttons remain enabled only for the export operation with **DONE** status.

- **Result**: Provides the result of an export operation. This result is available only for the export operations with **DONE** status. Following are the possible values:
  - **SUCCESS** : The export is successful
  - **FAILED** : The export is failed
  - **PARTIAL_SUCCESS** : The export is partially successful

Exporting Policy Projects with Dependencies

The Policy data export functionality explained in the above section allows you to export all the policy configurations with their dependencies. However, it does not export the dependencies of the selected Policy Projects.
The **Import & Export** page provides an option to export only the Policy Projects and Policies with their dependencies, without any MO configurations. The Policy Projects export is aligned with Policy services. You can export either all the Policy Projects or the projects of the selected services.

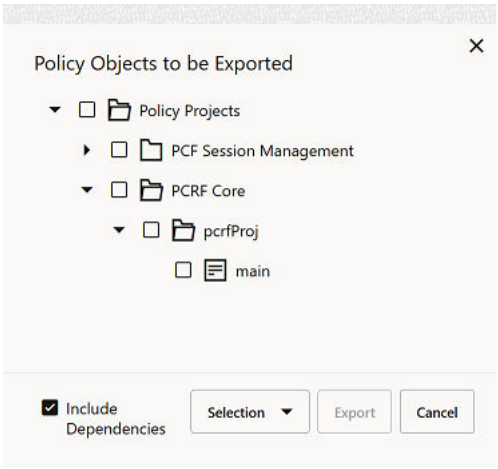
To export Policy Projects with dependencies:

1. On the **Import & Export** page, click 

.

   This opens the **Objects to be Exported** dialog box, displaying the list of Policy Projects for each service. The dialog box allows you to select the Policy Projects to be exported with an option to include dependencies. The following screen capture displays an illustration of the **Objects to be Exported** dialog box:

Figure 7-41 Objects to be Exported dialog box for Exporting Policies

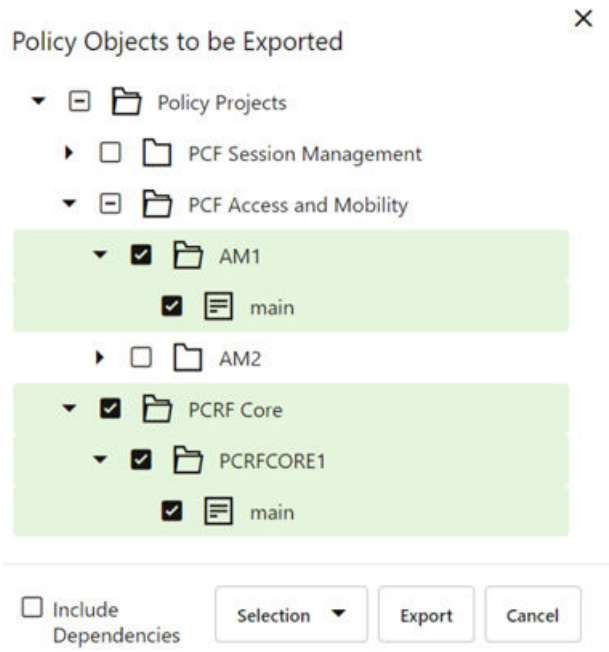


> **Note**
>
> Prior releases to Policy 23.1.0, Exporting of Managed objects and its dependencies is not supported for PCRF-CORE and PDS Services, this warning message "**Include Dependencies are not supported for PCRF-CORE & PDS Services**" is displayed.

2. In the dialog box, select the configurations that needs to be exported.

   Selecting a parent folder automatically selects all the child nodes. In case a folder is selected partially, then a partial selection sign is displayed on the parent folder as shown in the following illustration:

Figure 7-42 Objects to be Exported with Selected Policy Projects



> **Note**
>
> To select or deselect all the configurations, click **Selection** and perform the required operation.

3. Select the **Include Dependencies** check box to export all the dependant MOs of the selected policies.

   If you do not select the check box, then no dependent MOs for the selected policies get exported. It exports the single Policy Project ZIP file.

> **Note**
>
> The **Include Dependencies** functionality is not applicable for the PCRF Core and PDS services.

4. Click **Export**.

> **Note**
>
> Click **Cancel** to discard the export operation.

This starts the export of all the configurations related to the selected Policy Projects. A row is created in the export status table on the **Import & Export** page, displaying the export status with the following details:

- **Export ResourceId**: A new export resource ID is generated for each export operation. You can use this ID to get the export status.
- **Creation TimeStamp**: The timestamp of generation of **Export ResourceId**.
- **Progress (%)**: Shows the export progress in form of the percentage bar. The page auto refreshes the status until the progress reaches 100 percent.
- **Status**: The status of the export operation. The status can be any of the following:
  - **INIT**: The validation of policies is in progress and the export of the MOs is not yet started.
  - **IN PROGRESS**: The export is running.
  - **DONE**: The export is complete.
  - **FAILED** : The export is complete but errors exist in all the exported policies.
- **Actions**: Provides the buttons to download the following:
  - **Export configuration files in ZIP file format**: The exported configurations in ZIP file format. The ZIP file contains the MO data in JSON file format. You can download the exported data by clicking  under the **Action** column.

> **Note**
>
> If you have not selected the **Include Dependencies** check box, then the exported Zip file contains only single JSON file named PolicyProjects.json.

  - **Export report in TEXT file format**: The export report provides results for each JSON file present in the exported ZIP file. It also provides the reason for failure, in case the export of any of the configuration fails.

    You can download the export report by clicking  under the **Action** column.

    > **Note**
    >
    > The buttons remain enabled only for the export operation with **DONE** status.

- **Result**: Provides the result of an export operation. This result is available only for the operations with **DONE** status. Following are the possible value:
  - **SUCCESS** : The export is successful
  - **FAILED** : The export is failed.
  - **PARTIAL_SUCCESS** : The export is partially successful

7.8.1.2 Importing Policy Data

The import functionality allows you to import Policy data configurations. Using this functionality, you can import the same set of policy data to different PCF systems.
To import Policy data in JSON or ZIP file format:

1. From **Policy**, navigate to **Administration** and select **Import & Export**.

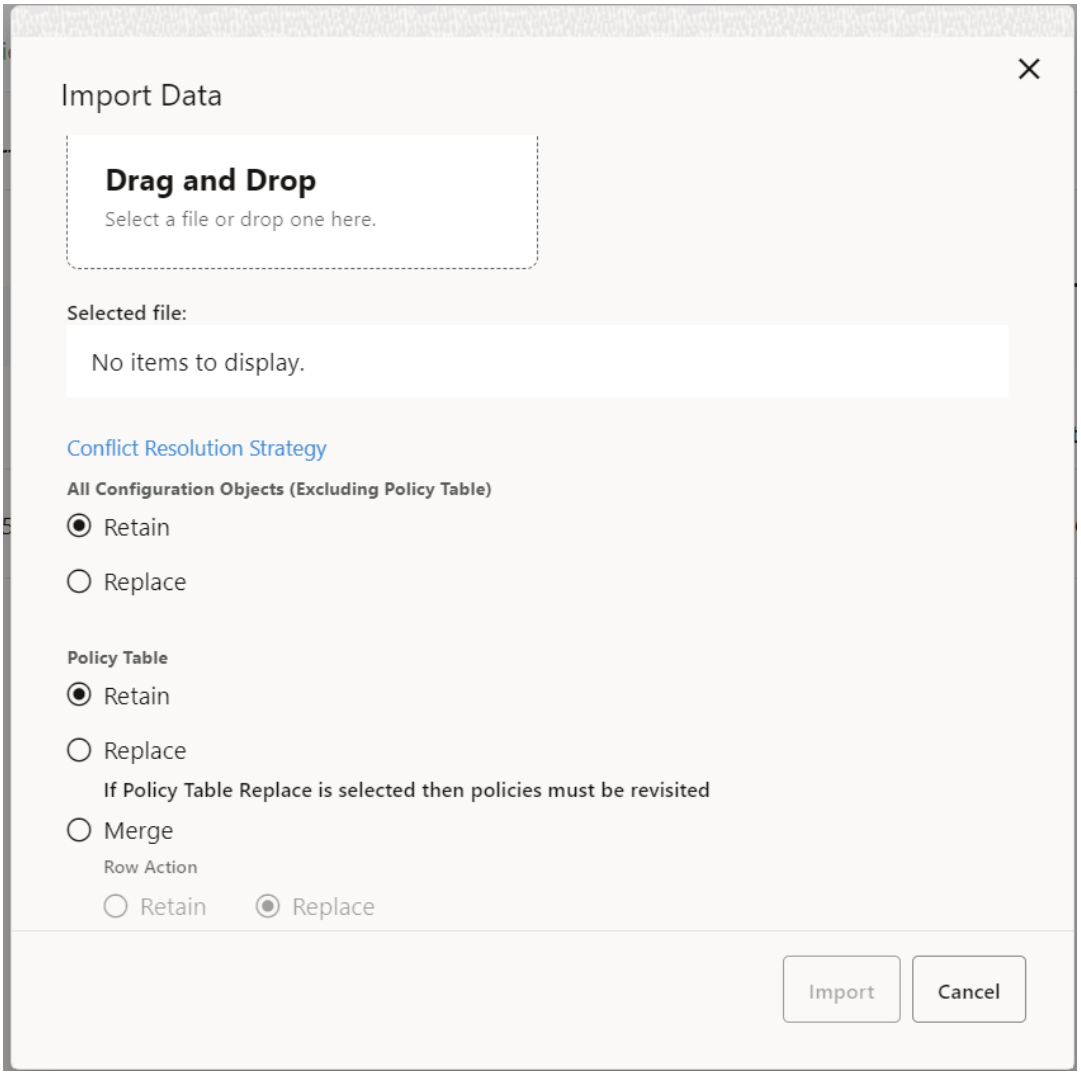
   This opens the **Import & Export** page, displaying the **Export** and **Import** tabs. By default, the **Export** tab remains selected.

2. Select the **Import** tab and click 

.

   This opens the **Import Data** dialog box.

Figure 7-43 Import Data



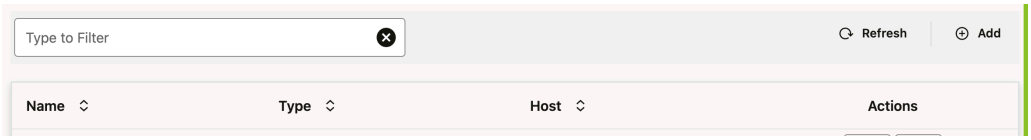
3. Upload the file in JSON or ZIP format by using the **Drag and Drop** button.

> **Note**
>
> You can not select the managed objects in the **Import Data** dialog box in Release 1.8.x.

4. Select any of the following options under **Conflict Resolution Strategy**.
   - **Attribute Configurations Screens**:

> **Note**
>
> This option is applicable to all the atttribute configuration screens such as **Data Sources** under **Data Source Configurations** or **Subscriber Activity Logging** under **Loggging Configurations** which have Add button at the top right corner along with a table. In such screens each row is considered as managed object.

Figure 7-44 Sample Attribute Configurations Screen



Conflict resolution:

- **Retain** : Ignores records already present in the database and does not overwrite. For each object in the import file, if the object already exists in the system, the import does not update the object with the configurations provided in the import file. If an object does not exists, then it is added to the system.
- **Replace** : For each object in the import file, if the object already exists in the system, the import replaces the object with the configuration provided in the import file. If an object does not exist, then it is added to the system.
  If an object existing in the system and is not present in the imported file, it is retained in the system.

- **Service Configuration Screens**:

> **Note**
>
> This option is applicable to all the service configuration screens such as **Session Management** under **Service Configurations** or **Settings** under **Diameter Configurations**, which have Edit button at the top right corner along with a table.

Figure 7-45 Sample Service Configurations Screen

Conflict resolution:

- **Retain** : Ignores the records and data which are getting imported. For each object in the import file, the imported record will be ignored.

- **Replace**: For each object in the import file, data to be imported will overwrites the existing data in the database, irrespective of whether the object is already present in the database or not.

5. Click **Import**.

> **Note**
>
> Click **Cancel** to discard the import operation.

This starts the import of configuration objects and their settings to the database. A row is created in the Import status table, displaying the import status with the following additional details:

- **Import ResourceId**: A new import resource ID is generated for each import operation. You can use this ID to get the import status.
- **Creation TimeStamp**: The timestamp of generation of **Import ResourceId**.
- **Progress (%)**: Shows the import progress in form of the percentage bar. The page auto refreshes the status until the progress reaches 100 percent.
- **Status**: The status of the import operation. The status can be any of the following:
  - **IN_PROGRESS**: The import is running.
  - **DONE**: The import is complete.
- **Actions**: Provides a button to download the import report in text format. This button gets enabled, once the status is **DONE**. The report provides results for each JSON file present in the imported ZIP file. You can download the import report by clicking  under the **Action** column.
- **Result**: Provides the result of an import operation. This result is available only for the operations with **DONE** status. Following are the possible value:
  - **SUCCESS** : The import is successful
  - **FAILED** : The import is failed
  - **PARTIAL_SUCCESS** : The import is partially successful

7.8.1.3 Using REST API for Policy Import & Export

Policy provides an option to perform the bulk export or import of Policy configurations and Policy Data using REST APIs. For more information about Bulk Import REST APIs, see "Bulk Import Export Controller" in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

7.9 Status and Query

This section describes how to retrieve status of Policy profile registration and query sessions using the Session Viewer page.

7.9.1 Session Viewer

The **Session Viewer** page displays the detailed session information of subscribers. You can use this page to enter query parameters to render session data for specific subscribers.

This procedure provides information about how to view session details and perform different operations for a single, multiple, or all subscribers of a specific session type.

To view a session:

1. From the navigation menu, click **Policy**, select **Status and Query**, and then click **Session Viewer**.
   This opens the Session Viewer page.
2. Select the session type for which you want to view the subscriber details. Select any of the following services from the **Session Type** drop-down list:
   - SM Policy Association
   - AM Policy Association
   - PA Policy Association
   - PCRF-Core Session
   - Binding Session
   - User Data
   - Usage Monitoring
   - UE Policy Association
   - ALL
3. From the **Identifier Type** drop-down menu, select the required identifier type.
   The following table lists the identifier types supported for each session type:

Table 7-159 Supported values for Identifier type

Session Type	Identifier Type
SM Policy Association	• SUPI • GPSI • IPV4 • IPV6 • POLICY_ASSOC_ID
AM Policy Association	• SUPI • GPSI • POLICY_ASSOC_ID • PEI
PA Policy Association	• SUPI • GPSI • IPV4 • IPV6 • POLICY_ASSOC_ID

Table 7-159 (Cont.) Supported values for Identifier type

Session Type	Identifier Type
PCRF-Core Session	• DIAMETER_SESSION_ID • IMSI • MSISDN • IPV4 • IPV6 **Note:** For the `PCRF-Core Session` type, the page displays the **Session Subtype** dropdown containing the session type that can be used. The dropdown values depends on the **Identifier Type** value. For more details of the Session Subtype field, see Step 4.
Binding Session	• SUPI • GPSI • IPV4 • IPV6 • POLICY_ASSOC_ID • MAC
User Data	• SUPI • GPSI
UE Policy Association	• SUPI • GPSI • PEI • POLICY_ASSOC_ID
Usage Monitoring	• SUPI • GPSI
ALL	• SUPI • GPSI **Note**: On selecting `ALL` as the session type, the following sessions are impacted: • SM session • PA session • UE session • AM session • Binding session • Diameter session • Usage Monitoring

4. <*Optional*> In case of **PCRF-Core Session** type, the page displays the **Session Subtype** dropdown containing the session type that can be used. The dropdown values depends on the **Identifier Type** value.
The following table lists the session subtypes supported for different identifier types:

Table 7-160 Supported values for Identifier type

Session Type	Identifier Type
• IMSI • MSISDN • IPV4 • IPV6	• All sessions • Gx Session • Rx Session • Sd Session

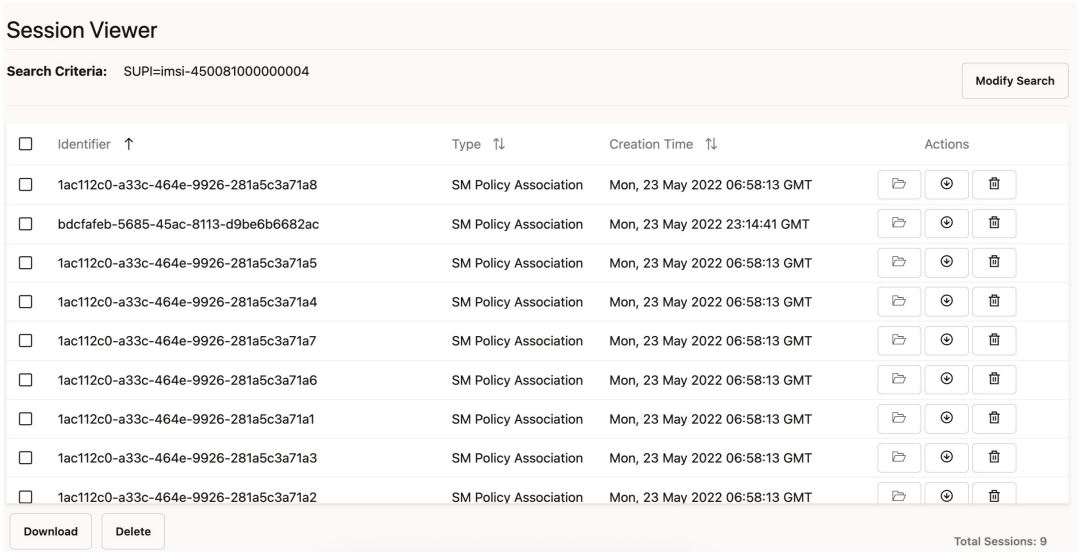
Table 7-160 (Cont.) Supported values for Identifier type

Session Type	Identifier Type
DIAMETER_SESSION_ID	• Gx Session • Rx Session • Sd Session

5. Enter the value in the **Identifier Value** field for the selected identifier type.
   For more information on identifier values, see Subscriber Identifiers in Subscriber Activity Logging.
6. Enable the **User Data** switch to get the user data from the Policy directory.
7. Click the **Query** button.
   The page displays the subscriber information according to the selected session type.

   The following image displays the query results for session type as `SM Policy Association` with `SUPI` as the identifier type:

Figure 7-46 Query Results on Session Viewer



The query results are displayed in tabular form with each row displaying the following details:

- **Identifier:** The selected session identifier.
- **Type:** Type of the session.
- **Creation Time:** The session creation time.
- **Action:** Provides the buttons to perform the following actions:
  - **View Session Details:** Click the

    icon to view the selected session details in the JSON editor.

- **Download Session Details:** Click the

  icon to download the selected file containing the details of the session on your system.

  > **Note**
  >
  > To download multiple session files in single instance, select the sessions and click the **Download** button at the bottom of the page.

- **Delete Sessions:** Click the

  icon to delete the selected session manually.

  > **Note**
  >
  > * To delete multiple sessions in a single instance, select the sessions and click the **Delete** button at the bottom of the page.
  > * For remote delete, enable
  >
  > 
  >
  >
  > box. When this option is enabled, Policy sends delete subscriptions to remote NFs when any previous subscriptions are enabled with remote NFs.
  >
  > > **Note**
  > >
  > > **Initiate Remote Session Cleanup** is applicable only for SM Service.
  > >
  > > In case of AM_Policy_Association and UE_Policy_Association, when a session is deleted, the corresponding context information in PDS is cleaned. If all the PDS context for UDR data are deleted, the subscription data in the UDR is also deleted.

> **Note**
>
> Use the **Modify Search** button to modify the search criteria.

Policy also provides an option to configure session viewer using REST APIs. For more information, see "Session Viewer" in *Cloud Native Core, Converged Policy REST Specification Guide*.

7.9.2 NF Status

This section provides information on NF status.

7.9.2.1 PCF Registration Profile

This page lists the PCF profile registered with NRF.

To make updates to any of the parameters of the PCF registration profile, perform the following steps:

1. Click **Edit** button.
2. Update the values of the required parameters.
3. Click **Save**.

To dowload the BSF profile, click **Download** button. A file named `pcfRegistrationProfile.json` is saved on your system.

7.9.2.2 NRF Status

On the NRF Status page, you can view the status of PCF and the NRFs deployed in the cluster.

For PCF

You can view the following details for PCF:

- **PCF status with NRF** - It shows whether the PCF instance is registered, suspended, or deregistered with NRF.
- **FQDN** - It shows the FQDN of the PCF registered with NRF.
- **Instance ID** - It shows the unique Instance ID of PCF registered with NRF.
- **Registered with** - It shows the FQDN of the NRF with which PCF is registered.
- **Registration Time** - It shows the time at which PCF registered with NRF.

If you want to view more details of the PCF instance such as its registration profile, click **View More Details**. It opens the **NF Registration Profile** page.

For NRF

You can view the following details for NRF:

- **Health Status** - This ribbon-styled badge shows the health status of the NRF instance. It could be in either healthy or unhealthy state.
- **Primary NRF** - The circular icon with the label P indicates that the NRF is primary.
- **Active Status** - The pulsating green circular icon shows that the NRF is currently active.
- **FQDN** - It shows the FQDN of the NRF.
- **Priority** - It shows the priority of the NRF instances. An NRF instance with priority 1 is treated as primary NRF.
- **Last Connected Time** - It shows the time when PCF last connected with primary NRF.
- **Last Status Change Time** - It shows the time when the NRF status changed.

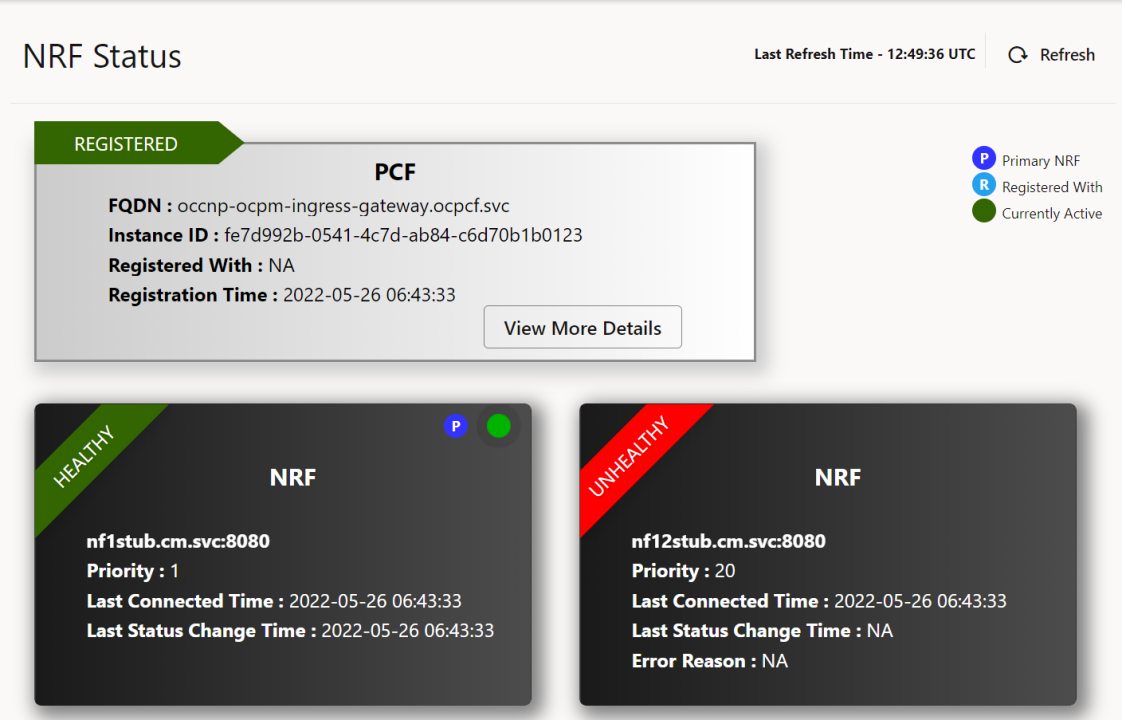
When the status of an NRF instance changes from healthy to unhealthy, the error reason is also displayed on the page.

To reload the data on the page at any time, click the **Refresh** button.

Based on when the data refreshes, the Last Refresh Time on the page is also updated.

The number of NRFs displayed on the page are dynamic, and get updated according to the NRFs configured in the network.

Figure 7-47 NRF Status



> **Note**
>
> Currently, the page is known to display the status of up to 8 NRF Instances.

In case of any network error, the page displays an error - **Unable to load NRF data**.

7.9.2.3 Discovered NF Instances

This page displays the list of discovered NF Instances.

On the **Discovered NF Instances** page, you can view the autonomously discovered NF profiles for BSF, CHF, UDR, and NWDAF.

Based on the value you select from the drop-down list, the following details are populated on the page:

- **NF Instance ID -** It shows the unique instance ID of the discovered NF.
- **FQDN** - It shows the FQDN of the discovered NF.

- **Priority** - It shows the priority assigned to the NF.
- **Status** - It shows whether the NF is registered or suspended.
- **View Details** - Click to view the NF profile.

If no profile is discovered or in case of any network error, the page shows "**No data to display**" along with 404-Not Found.

7.10 Overload and Congestion Control Configurations

This section describes how to perform overload and congestion control configurations.

To use the Error Code Profiles page to create and manage error code profiles in Overload Control Configurations for Diameter Gateway and SBI interface, see *SBI Ingress Error Code Profiles Collection* section.

> **Note**
>
> When overload control feature is enabled, it should be enabled for both Diameter gateway and SBI interface. The overload control manager needs data from both Ingress Gateway and Diameter Gateway to determine the overall load on BSF management. It is not possible to enable overload control for Diameter Gateway and disable the same in Ingress Gateway.

7.10.1 Diameter

This section describes configuring the Load Shedding Profiles and the Message Priority Profiles for Diameter Gateway.

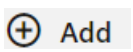
To use the Error Code Profiles page to create and manage error code profiles in Overload Control Configurations for Diameter Connector, see SBI Ingress Error Code Profiles Collection.

7.10.1.1 Load Shedding Profiles

This procedure provides information about how to create and manage load shedding profiles in Diameter Configurations.

The Load Shedding Profiles page allows you to create new and manage existing load shedding profiles. The page displays the list of defined profiles and provides the options to import and export data as well.

To configure Load shedding profiles, perform the following steps:

1. From the navigation menu, under **Policy**, click **Diameter Configurations**, and select **Load Shedding Profiles**.
   This opens the Load Shedding Profiles page.
2. Click 

   .
   This opens the Create Load Shedding Profiles page.
3. Enter values for the available input fields, described in the following table:

Table 7-161 Load Shedding Profiles Configurations

Field Name	Description
Name	Unique name of the load shedding profile.
Scheme	Allows to configure the discard policy based on • Priority: to discard messages based on priority range • Priority and Percentage: to discard messages based on priority range and percentage for each range
Type	Defines the type of load shedding profile. You can select any of the following values from the drop-down list: • Congestion Control • Overload Control

To add load shedding rules for the profile type congestion control, perform the following steps:

a. Under **Load Shedding Rules**, click Add.
This opens the **Add Load Shedding Rules** dialog box.

b. Enter values for the available input fields, described in the following table:

Table 7-162 Load Shedding Rules Configurations When the Selected Scheme is "Priority"

Field Name	Description
State	This field appears when the **Type** of load shedding profile is `Congestion Control`. Specifies the type of state for which the rule is being defined. Select any of the following values using the drop-down: • Danger of Congestion • Congested
Discard Priority	This field appears when the **Scheme** of load shedding profiles is `Priority`. Specifies the discard priority for the load shedding rule. The discard priority value can be a number from 0 to 15. Any request message with equal or greater priority is rejected.

Table 7-162 (Cont.) Load Shedding Rules Configurations When the Selected Scheme is "Priority"

Field Name	Description
Answer with Result Code	Specifies the result code, returned in the answer response, when request message is rejected as part of congestion control. Users can select any of the following values from the drop-down menu: • DIAMETER_TOO_BUSY • DIAMETER_UNABLE_TO_COMPLY • DIAMETER_UNABLE_TO_DELIVER • EXPERIMENTAL_RESULT_CODE **Note**: When the EXPERIMENTAL_RESULT_CODE value is selected, the following two fields are populated on the page: • **Result Code**: Enter a custom result code. • **Vendor ID**: Enter a valid value to specify vendor ID.

Table 7-163 Load Shedding Rules Configurations When the Selected Scheme is "Priority and Percentage"

Field Name	Description
State	This field appears when the **Type** of load shedding profile is `Congestion Control`. Specifies the type of state for which the rule is being defined. Select any of the following values using the drop-down: • Danger of Congestion • Congested
Discard Priority Percentage	Allows to configure the Discard Priority Percentage as explained in the following step.

c. To configure discard Priority Percentage, perform the following steps:

i. Under **Discard Priority Percentage**, click ⊕ Add .
This opens the **Add Discard Priority Percentage** dialog box.

Enter values for the available input fields, described in the following table:

Table 7-164 Adding Discard Priority Percentage

Field Name	Description
Priority Range	Specifies the discard priority range for the load shedding rule. The discard priority value can be a number from 0 to 15. Any request message with equal or greater priority will be rejected based on the percentage discard configured. **Note**: For Priority added as percentage type, when a single digit number is provided in the priority range then those many numbers of priority messages will be rejected. For example: If Priority is 9, only 9 priority messages will be rejected.
Discard Percentage	Specifies the discard percentage for the specified priority range.
Answer with Result Code	Specifies the result code, returned in the answer response, when request message is rejected as part of congestion control. Users can select any of the following values from the drop-down menu: • DIAMETER_TOO_BUSY • DIAMETER_UNABLE_TO_COMPLY • DIAMETER_UNABLE_TO_DELIVER • EXPERIMENTAL_RESULT_CODE **Note**: When the EXPERIMENTAL_RESULT_CODE value is selected, the following two fields are populated on the page: • **Result Code**: Enter a custom result code. • **Vendor ID**: Enter a valid value to specify vendor ID.

ii. Click **Save** to save the discard priority percentage.

d. Click **Save** to save the load shedding rule.

OR

To add load shedding rules for the profile type overload control, perform the following steps:

a. Under **Load Shedding Rules**, click ⊕ Add .
This opens the **Add Load Shedding Rules** dialog box.

b. Enter values for the available input fields, described in the following table:

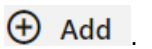
Table 7-165 Load Shedding Rules Configurations When the Selected Scheme is "Priority"

Field Name	Description
Level	Specifies the name of the level. The name specified in this parameter must match the level name in Ingress Gateway's ocdiscardpolicies. Select any of the following values using the drop-down list: • L1 • L2 • L3 **Note**: Any existing L4 level data will be removed, as L4 is not supported.
Discard Priority	Specifies the discard priority for the load shedding rule. The discard priority value can be a number from 0 to 15. Any request message with equal or greater priority is rejected.
Answer with Result Code	Specifies the result code, returned in the answer response, when request message is rejected as part of overload control. Users can select any of the following values from the drop-down menu: • DIAMETER_TOO_BUSY • DIAMETER_UNABLE_TO_COMPLY • DIAMETER_UNABLE_TO_DELIVER • EXPERIMENTAL_RESULT_CODE **Note**: When the EXPERIMENTAL_RESULT_CODE value is selected, the following two fields are populated on the page: • **Result Code**: Enter a custom result code. • **Vendor ID**: Enter a valid value to specify vendor ID.

Table 7-166 Load Shedding Rules Configurations When the Selected Scheme is "Priority and Percentage"

Field Name	Description
Level	Specifies the name of the level. The name specified in this parameter must match the level name in Ingress Gateway's ocdiscardpolicies. Select any of the following values using the drop-down list: • L1 • L2 • L3 **Note**: Any existing L4 level data will be removed, as L4 is not supported.
Discard Priority Percentage	Configure the discard priority percentage as explained in the following step.

c. To configure Discard Priority Percentage, perform the following steps:

i. Under **Discard Priority Percentage**, click 

.

This opens the **Add Discard Priority Percentage** dialog box.

Enter values for the available input fields, described in the following table:

Table 7-167 Adding Discard Priority Percentage

Field Name	Description
Priority Range	Specifies the discard priority range for the load shedding rule. The discard priority value can be a number from 0 to 15. Any request message with equal or greater priority will be rejected based on the percentage discard configured. **Note**: For Priority and Percentage type, when we give single number in the priority range, for example 9, only 9 priority messages will be rejected.
Discard Percentage	Specifies the discard percentage for the specified priority range.
Answer with Result Code	Specifies the result code, returned in the answer response, when request message is rejected as part of congestion control. Users can select any of the following values from the drop-down menu: • DIAMETER_TOO_BUSY • DIAMETER_UNABLE_TO_COMPLY • DIAMETER_UNABLE_TO_DELIVER • EXPERIMENTAL_RESULT_CODE **Note**: When the EXPERIMENTAL_RESULT_CODE value is selected, the following two fields are populated on the page: • **Result Code**: Enter a custom result code. • **Vendor ID**: Enter a valid value to specify vendor ID.

ii. Click **Save** to save the discard priority percentage.

d. Click **Save** to save the load shedding rule.

4. Click **Save** to save the load shedding profile.
To discard the changes, click **Cancel**

The value gets listed on the Load Shedding Profiles page. Use or available under the **Actions** column to update or delete the profile.

> **Note**
>
> While Diameter Gateway enters overload situation, messages only from network peer will be considered for load shedding. That is, messages from backed peers will not be considered for load shedding.
>
> For example, during overload Sy-SLR / Rx-RAR will not be considered for load shedding as they are generated by backend microservices.

Importing Load Shedding Profiles

To import load shedding profiles, perform the following steps:

1. Click Import .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

Exporting Load Shedding Profiles

To export load shedding profiles, click **Export**. A `json` file is saved to your device.

7.10.1.2 Message Priority Profiles

This procedure provides information about how to create and manage message priority profiles in Diameter Configurations.

The Message Priority Profiles page allows you to create new and manage existing message priority profiles. The page displays the list of defined profiles and provides the options to import and export data as well.

To configure Message Priority profiles, perform the following steps:

1. From the navigation menu, under **Policy**, click **Diameter Configurations**, and select **Message Priority Profiles**.
   This opens the Message Priority Profiles page.
2. Click Add .
   This opens the Create Message Priority Profiles page.
3. Enter values for the available input fields, described in the following table:

Table 7-168 Message Priority Profiles Configurations

Field Name	Description
Name	Unique name of the message priority profile.

To add message priority rules for the profile, perform the following steps:

a. Under **Message Priority Rules**, click Add .
   This opens the **Add Message Priority Rules** dialog box.
b. Enter values for the available input fields, described in the following table:

Table 7-169 Message Priority Rules Configurations

Field Name	Description
Name	Specifies the unique name of the message priority rule.

Table 7-169 (Cont.) Message Priority Rules Configurations

Field Name	Description
Message Priority	Specifies the priority assigned to the message. It can be a number from 0 to 15. **Note**: The default message priority is 10, if no rule is available. For setting a different default message priority, a rule must be added with lowest rule priority(higher number) to match all the messages(application set to "any", message type set to "any"). This will ensure that, this rule will be run at the last i.e., when no other rules match.
Rule Priority	Specifies the priority assigned to the message priority rule. The range for this field is 0 to 65535. **Note:** During Import and Export, if the old data has rule priority value greater than the specified range, then the Rule Priority might be impacted.
Use DRMP Priority	When this switch is enabled, the priority for the message rule is assigned from DRMP AVP.
Conditions	
Application	Specifies the type of application. Users can select any of the following values from the drop-down: • Gx • Sd • Rx • Sy • Any
Message	Specifies the type of message for the selected application. The supported message values for each application type are as follow: • For Gx application, choose a value from CCR, RAR, and Any. • For Sd application, choose CCR. • For Rx application, choose a value from AAR, ASR, RAR, STR, and Any. • For Sy application, choose a value from Sy-SLR, STR, Sy-SNR, and Any. • For application type selected as Any, choose a value from the drop-down list of **Message**.

c. To add pre-defined AVP conditions, click ⊕ Add under **Pre Defined AVP Conditions**. On the **Add Pre Defined AVP Conditions** dialog box, select Name and enter values as described in the following table:

Table 7-170 Pre Defined AVP Conditions Configurations

Name	Values
Called-Station-Id	This AVP can be used only when the application type is specified as Gx and Rx. Users can enter multiple comma-separated values. **Note**: CNC Policy supports wildcard format for this AVP.
CC-Request-Type	This AVP can be used for Gx application with message specified as CCR. You can select any of the following valid values from the drop-down list: • INITIAL_REQUEST • UPDATE_REQUEST • TERMINATION_REQUEST • EVENT_REQUEST
Rx-Request-Type	This AVP can be used for Rx application with message specified as AAR. You can select any of the following valid values from the drop-down list: • INITIAL_REQUEST • UPDATE_REQUEST • PCSCF_RESTORATION_REQUEST
Service-URN	This AVP indicates that an AF session is used for emergency traffic. It is of type OctetString. Examples: "sos", "sos.fire", "sos.police" and "sos.ambulance". **Note**: CNC Policy supports wildcard format for this AVP.
MPS-Identifier	This AVP indicates that an AF session relates to an MPS session and contains the national variant for MPS service name. It is of type OctetString. Example: NGN GETS
MCPTT-Identifier	This AVP includes either one of the namespace values used for MCPTT and may include the name of the MCPTT service provider. It is of type OctetString.
MCVideo-Identifier	This AVP includes the name of the MCVideo service provider. It is of type OctetString.
Reservation-Priority	This AVP is of type Enumerated and is specified in an AA-Request as the main AVP to associate a priority with a resource reservation or modification request. You can specify a value from 0 to 7 for this AVP.
SN-Request-Type	This AVP informs the PCRF about the type of the Spending-Status-Notification-Request (SNR).You can select any of the following valid values from the drop-down list: • ABORT_SESSION_REQUEST • NORMAL_REQUEST

Click **Save** to save the pre-defined AVP conditions for the message priority rule.

d. Click **Save** to save the message priority rule.

4. Click **Save** to save the message priority profile.
   To discard the changes, click **Cancel**

   The value gets listed on the Message Priority Profiles page. Use  or  available under the **Actions** column to update or delete the profile.

Importing Load Shedding Profiles

To import message priority profiles, perform the following steps:

1. Click  .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

Exporting Message Priority Profiles

To export load shedding profiles, click **Export**. A `json` file is saved to your device.

7.10.2 SBI

This section describes the Rate Limiting and overload control for SBI.

To use the Error Code Profiles page to create and manage error code profiles in Overload Control Configurations for SBI interface, see SBI Ingress Error Code Profiles Collection.

7.10.2.1 Rate Limiting

This section describes Rate Limiting Policy and Route Level Mapping for SBI.

> **Note**
>
> As of now if the system requires overload configuration for both rate limiting and failure count, it is suggested to do these configuration using API's only. For more details, see *Rate Limiting at Ingress Gateway* and *Failure Count at Ingress Gateway* sections in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*

7.10.2.1.1 Rate Limiting Policy

This procedure provides information about how to use the Rate Limiting Policy page to manage rate limiting policies for overload control on SBI interface.

To configure rate limiting policy, perform the following steps:

1. From the navigation menu, under **Policy**, click **Overload Control Configurations**, select **SBI**, then select **Rate Limiting**, and then select **Rate Limiting Policy**.
   This opens the Rate Limiting Policy page.
2. Click Edit.
   This opens the Edit Rate Limiting Policy page.
3. Enter values for the available input fields as described in the following table:

Table 7-171 Rate Limiting Policy Configurations

Field Name	Description
Enable Rate Limiting	Specifies whether to enable or disable rate limiting.
Sampling Period (in milliseconds)	Specifies the time frame for each cycle of rate limiting per service. Its default value is 200 ms.

4. Under **Rate Limit Policy**, click Add .

   This opens the Add Rate Limit Policy dialog box.

5. Enter values for the available input fields as described in the following table:

Table 7-172 Rate Limit Policy Configurations

Field Name	Description
Name	Specifies the name of the rate limit policy that is further used to determine a mapping between route and discard policy name per route.
Discard Priority	Specifies the discard priority for the rate limiting policy. Any request with message priority higher in value than the discard priority is rejected.
Action	Specifies the action taken when when requests are discarded. Currently, the only supported value is RejectWithErrorCode.
Scheme	Specifies the scheme for applying rate limiting. Currently, the only supported value is PriorityBased.
Error Code Profile	Specifies the list of error code profiles configured on the **Error Code Profiles** page.

6. Click **Save** to save the rate limit policy. The value gets listed under the **Rate Limit Policy** group.
   To discard the changes, click **Cancel**.

   Use or available under the **Actions** column to update or delete any given policy.

7. Click **Save** to save the rate limiting policy.
   To discard the changes, click **Cancel**.

7.10.2.2 Overload Control

This section describes the Discard Policy Mapping and Discard Policy for SBI interface.

7.10.2.2.1 Discard Policy Mapping

This procedure provides information about how to use the Discard Policy Mapping page to manage discard policy mapping in Overload Control Configurations for SBI interface.

To configure Discard Policy Mapping, perform the following steps:

1. From the navigation menu, under **Policy**, click **Overload Control Configurations**, select **SBI**, then select **Overload Control**, and then select **Discard Policy Mapping**.
   This opens the Discard Policy Mapping page.
2. Click Edit.

This opens the Edit Discard Policy Mapping page.

3. Enter values for the available input fields as described in the following table:

Table 7-173 Discard Policy Mapping Configurations

Field Name	Description
Enable Overload Control	Specifies whether to enable or disable overload control.
Sampling Period (in milliseconds)	Specifies the time frame for each cycle of overload control per service. Its default value is 200 ms.

4. Under **Mappings**, click Add .

   This opens the Add Mappings dialog box.

5. Enter values for the available input fields as described in the following table:

Table 7-174 Mappings Configurations

Field Name	Description
Service Name	Specifies the name of the microservice that is further used to determine a mapping between service and discard policy name per service.
Policy Name	Specifies the name of the discard policy that is used to determine a mapping between service and discard policy name per service. The drop-down list shows the policies configured using the Discard Policy page.

6. Click **Save** to save the mappings.
   To discard the changes, click **Cancel**

   The value gets listed under the **Mappings** group on the Discard Policy Mapping page. Use or available under the **Actions** column to update or delete the mappings.

7. Click **Save** to save the discard policy mapping.
   To discard the changes, click **Cancel**.

7.10.2.2.2 Discard Policy

This procedure provides information about how to use the Discard Policy page to manage discard policies for overload control for SBI interface.

To configure discard policy, perform the following steps:

1. From the navigation menu, under **Policy**, click **Overload Control Configurations**, select **SBI**, then select **Overload Control**, and then select **Discard Policy**.
   This opens the Overload Control Discard Policy page.
2. Click Edit.
   This opens the Edit Overload Control Discard Policy page.
3. Click Add .

   This opens the Add Discard Policies page.
4. Enter values for the available input fields as described in the following table:

Table 7-175 Discard Policy Configurations

Field Name	Description
Name	Specifies the unique name of the discard policy.
Scheme	Specifies the criteria of dropping requests for a microservice. It could be either priority based or percentage based. If you select the value as Priority Based, configure the values of the parameters under Priority Based Policies. For Percentage Based scheme, configure the parameters under Percentage Based Policies.

To add priority based policies, perform the following steps:

a. Under **Priority Based Policies**, click Add.
This opens the **Add Priority Based Policies** dialog box.

b. Enter values for the available input fields as described in the following table:

Table 7-176 Priority Based Policies Configurations

Field Name	Description
Level	Specifies the name of the level. The name specified in this parameter must match the level name in Ingress Gateway's ocdiscardpolicies. Select any of the following values using the drop-down list: • L1 (Load Level 1) • L2 (Load Level 2) • L3 (Load Level 3)
Discard Priority	Specify the discard priority for the discard policy rule. Any request message with equal or lower message priority is rejected. **Note** 1 is considered as the highest message priority.
Error Code Profile	Select an error code profile from the drop-down list. It displays the list of error profiles configured using the Error Code Profile page.
Action	Specifies the action taken when selected requests are rejected. Currently, it only supports the action to reject requests based on error code.

OR

Table 7-177 Percentage Based Policies Configurations

Field Name	Description
Level	Specifies the name of the level. The name specified in this parameter must match the level name in Ingress Gateway's ocdiscardpolicies. Select any of the following values using the drop-down list: • L1 (Load Level 1) • L2 (Load Level 2) • L3 (Load Level 3)
Discard Percentage	Specify the discard percentage for the policy rule. The specified percentage of the calculated rate for service in previous sampling period is discarded in current sampling period.
Error Code Profile	Select an error code profile from the drop-down list. It displays the list of error profiles configured using the Error Code Profile page.
Action	Specifies the action taken when selected requests are rejected. Currently, it only supports the action to reject requests based on error code.

c. Click **Save** to save the discard policy.
To discard the changes, click **Cancel**.

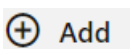
5. Click **Save** to save the overload control discard policy.
To discard the changes, click **Cancel**

The value gets listed on the Overload Control Discard Policy page. Use  or  available under the **Actions** column to update or delete any given policy.

7.10.2.3 Route Level Mapping

This procedure provides information about how to use the Route Level Mapping page to manage route level mapping for overload control on SBI interface.

To configure route level mapping, perform the following steps:

1. From the navigation menu, under **Policy**, click **Overload Control Configurations**, select **SBI**, then select **Rate Limiting**, and then select **Route Level Mapping**.
This opens the Route Level Mapping page.
2. Click Edit.
This opens the Edit Route Level Mapping page.
3. Under **Route Configuration**, click 

.

This opens the Add Route Configuration dialog box.
4. Enter values for the available input fields as described in the following table:

Table 7-178 Rate Limiting Policy Configurations

Field Name	Description
Id	Specifies the list of route IDs available for Policy.

5. Under **Rate Limiting**, click Add .

   This opens the Add Method dialog box.

6. Enter values for the available input fields as described in the following table:

Table 7-179 Method Configurations

Field Name	Description
Http Method	Specifies the HTTP method. Depending on the value select for **Id**, you can select any of the following values from the drop-down list: • POST • PUT • GET • DELETE • PATCH
Message Rate (per sampling period)	Specifies the message rate per sampling period for a given method.
Rate Limit Policy	Select a rate limit policy from the drop-down list. It displays the list of rate limit policies configured using the Rate Limiting Policy page.

7. Click **Save** to save the method. The value gets listed under the **Rate Limiting** group.
   To discard the changes, click **Cancel**.

   Use or available under the **Actions** column to update or delete any given policy.

8. Click **Save** to save the route configuration.
   To discard the changes, click **Cancel**.

9. Click **Save** to save the route level mapping.
   To discard the changes, click **Cancel**.

7.10.2.4 Failure Count

> **Note**
>
> Currently, all the configurations for failure count can be performed only using REST APIs.
>
> Only the failure count threshold can be configured using CNC Console. Configuring the error codes and mapping the respective routes must be done only through REST APIs.
>
> Also, if the system requires overload configuration for both rate limiting and failure count, it is suggested to perform these configuration using API's only.
>
> For more details, see *Rate Limiting* and *Failure Count at Ingress gateway* sections in *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.
>
> From 24.1.0 release, you can configure failure count using CNC Console. For more details, see IGW .

7.10.3 Congestion Control

7.10.3.1 Settings

The **Settings** page displays the general configurations related to the Congestion Control. The page allows you to edit the configurations.

To edit settings:

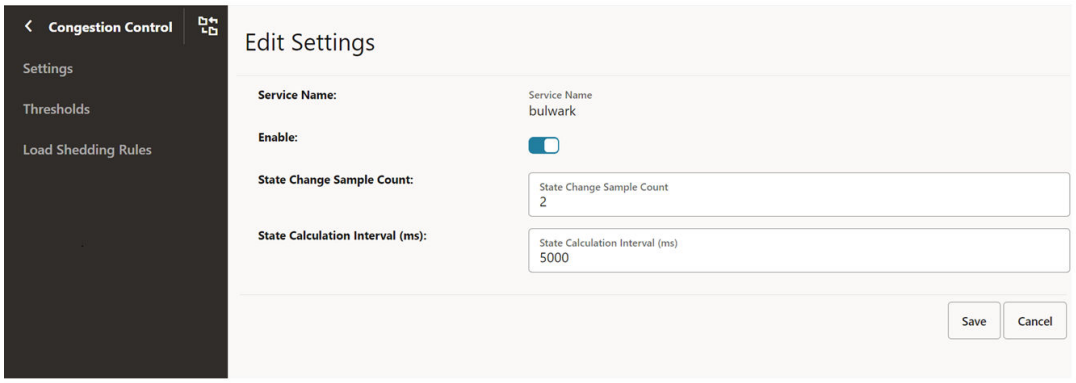
1. From the navigation menu under **Policy**, click **Overload and Congestion Control** and select **Settings**.
   This opens the **Settings** page. The page displays the existing configurations.
2. Click 

   .
   This opens the **Edit Settings** page.
3. Enter values for the available input fields as described in the following table:

Table 7-180 Edit Configurations

Field Name	Description
Service Name	Specifies the Policy service name. You can check the default configuration for the all fields and edit as necessary.
Enable	Specifies whether to enable or disable Congestion Control for the selected Policy service. **Default value**: False
State Change Sample Count	This specifies after how many continuous intervals, the pod state can be changed. This count can range from 1 to 2147483647. **Default Value**: Bulwark Service: 2 Binding, PDS, Usage Monitoring, SM Service: 5
State Calculation Interval (in milliseconds)	This specifies that after this time duration or interval, the pod congestion state will be re-verified. This inverval in milliseconds can range from 50 to 2147483647. **Default Value** Bulwark Service: 5000 Binding , PDS, Usage Monitoring, SM Service: 200

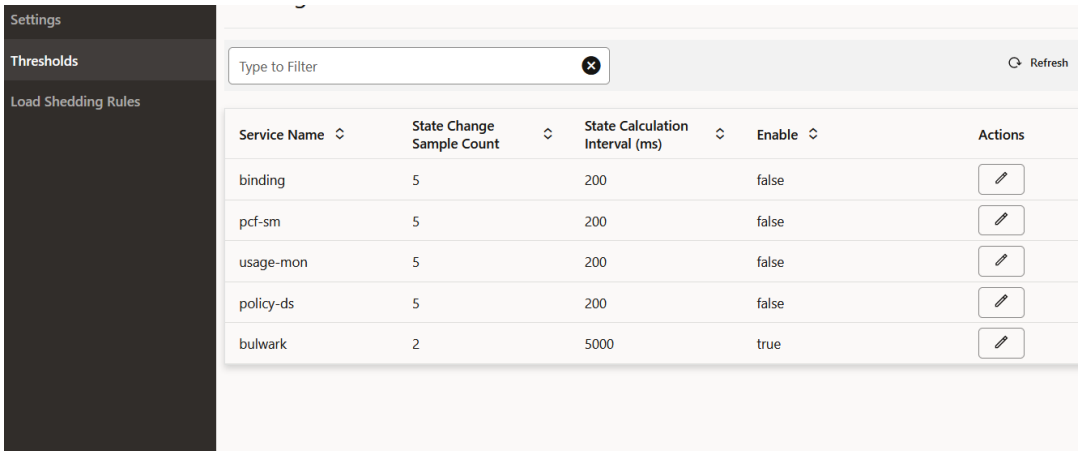
Figure 7-48 Edit Congestion Control Settings for Bulwark Service



4. Click **Save** to save the congestion control configuration.
   To discard the changes, click **Cancel**.

   The value gets listed on the Settings page.

Figure 7-49 Congestion Control Settings



7.10.3.2 Thresholds

Policy allows you to configure the congestion control threshold values for a Policy service using a threshold profile. You can either use a default that is set during PCF deployment or create a new profile using CNC Console. You must activate one of the profiles to use the threshold values. At any time, you can activate only one profile. To open **Congestion Control Thresholds** page:

1. From the navigation menu under **Policy** navigate to **Overload and Congestion Control**, select **Congestion Control** and select **Thresholds**.

   This opens the **Congestion Control Thresholds** page. The page shows the existing configurations for a specific Policy service. You can add a new threshold profile configuration or copy the existing profile and modify it.

2. Select from the drop down list fields, as described in the following table:

Table 7-181 Thresholds Configuration

Field Name	Description
Service Name	Provides a drop down lists of Policy services. You can select the service for which the congestion control thresholds are to be configured.
Threshold Profiles	Provides a drop down lists of all the configured threshold profiles. A default congestion control threshold profile is provided during PCF deployment. You cannot either edit or delete this profile. If the selected profile is already active, you can see **Active** button in green color next to the profile name.

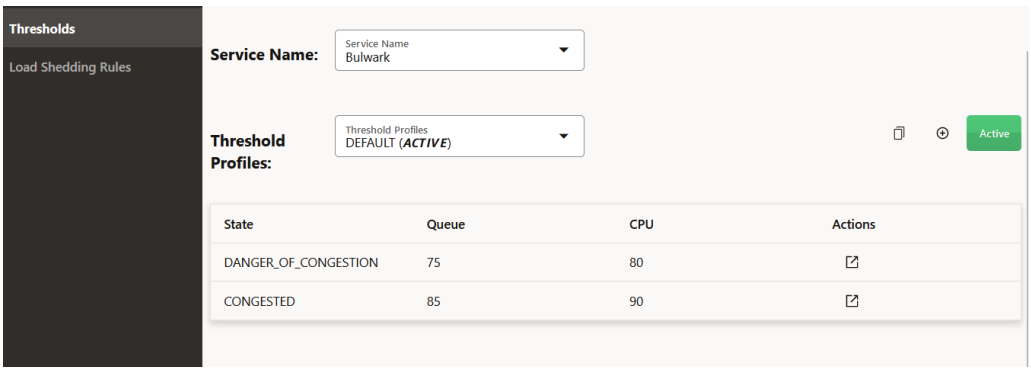
3. Click **Go To Active** button. The current active profile from the **Threshold Profile** drop down list is selected.
4. Click Add.
   This opens the **Create Profile** window and allows you to create a new profile.
   a. Enter values for the available input fields, described in the following table:

Table 7-182 Create Profile Configurations

Field Name	Description
Service Name	Specifies the selected PCF service for which a new threshold profile is to be created.
Name of the profile	Provide a unique name for the threshold profile to be created. The name can only contain the characters A–Z, a–z, 0–9, period (.), hyphen (-), and underline (_). The maximum length is 255 characters.

   b. Click **Create** to create the threshold profile.
   To discard the changes, click **Cancel**.
   c. The value gets listed on the Congestion Control Thresholds page.
   By default, the newly created profile has the same values as the default profile configuration values.

Figure 7-50 Bulwark Default Threshold



d. Click  to view the Queue and CPU values for the various congested states.

e. Click  to edit each of the various congested states **Queue** and **CPU** values. This opens the **Edit <Congestion States>** page, where <Congestion States> will be different for different services.
The Bulwark pod's can be in following Congestion states:

Table 7-183 Bulwark Congestion States

Congestion States	CPU Percentage	Queue Percentage
DANGER_OF_CONGESTION (DOC)	80	75
CONGESTED	90	85

> **Note**
>
> If user either adds or updates the Threshold configurations for Bulwark service, then Congestion Control feature gets disabled. The user will have to enable the congestion control feature again for Bulwark service using the **Settings** menu in CNC Console, and the new/updated Threshold configurations will be applied.

The Binding service pod's can be in following Congestion states:

Table 7-184 Binding Service Congestion States

Congestion States	CPU Percentage	Queue (Pending Requests)
DANGER_OF_CONGESTION (DOC)	70	30
CONGESTION_L1	75	36
CONGESTION_L2	80	42
CONGESTED	85	48

The SM Service pods can be in following Congestion states:

Table 7-185 SM Services Pod Congestion States

Congestion States	CPU Percentage	Queue (Pending Requests)
DANGER_OF_CONGESTION (DOC)	77	140
CONGESTION_L1	78	160
CONGESTION_L2	79	180
CONGESTED	80	200

The Policy service pod's can be in following Congestion states:

Table 7-186 Usage Monitoring and PDS Service Congestion States

Congestion States	CPU Percentage	Queue (Pending Requests)
DANGER_OF_CONGESTION (DOC)	65	50
CONGESTION_L1	70	100
CONGESTION_L2	75	150
CONGESTED	80	200

f. Enter the following information:

Table 7-187 Edit <Congestion State> Fields

Field Name	Description
Service Name	Specifies the selected service for which threshold profile configurations are to be edited.
Profile Name	Specifies the selected threshold profile for which DOC type queue and CPU values are to be edited.
Queue	Specifies the Queue percentage for Bulwark and Queue (pending requests) for all other Policy services across all the congestion levels. For Bulwark the range is 1 to 100 and for other services the range is between 1 to 2147483647.
CPU	Specifies the CPU percentage across all the congestion levels. The number can be between 1 to 100.

g. Click **Save** to save the configurations.
To discard the changes, click **Cancel**.

The updated value gets listed on the Congestion Control Thresholds page.

5. Click  . This opens the **Copy Profile** window to copy an existing profile.
While using the copy profile option, whichever profile is selected in the **Threshold Profile** dropdown, the values are copied from that profile.

6. Click  . This deletes the selected threshold profile. You cannot delete the default system provided threshold profile.

7.10.3.3 Load Shedding Rules

The Congestion Load Shedding Rules page allows you to create new and manage existing congestion load shedding rules.

To configure Congestion Load Shedding rules, perform the following steps:

1. From the navigation menu under **Policy** navigate to **Overload and Congestion Control**, select **Congestion Control** and then select **Load Shedding Rules**.
This opens the Congestion Load Shedding Rules page. The page shows the existing configurations for a specific Policy service. You can add a new congestion control load shedding rule or copy the existing rule and modify it.

2. Select from the drop down list fields, as described in the following table:

Table 7-188 Congestion Load Shedding Rules

Field Name	Description
Service Name	Provides a drop down lists of Policy services. You can select the service for which the congestion load shedding rules are to be configured.
Load Shedding Rules	Provides a drop down lists of all the configured load shedding rules. A default load shedding rule is provided during PCF deployment. You cannot either edit or delete this rule. If the selected profile is already active, you can see **`Active`** button in green color next to the profile name.

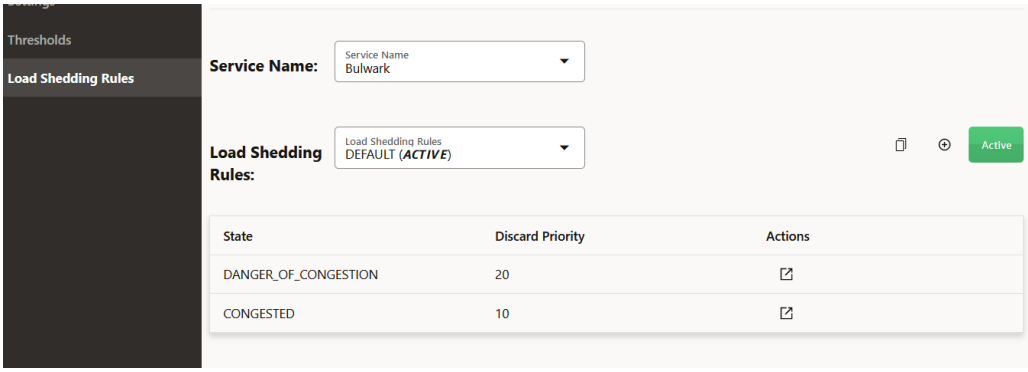
3. Click **Go To Active** button. The current ACTIVE rule from the **Load Shedding Rules** drop down list is selected.
4. Click ⊕ Add .
   This opens the **Create rule** window and allows you to create a new rule.
   a. Enter values for the available input fields, described in the following table:

Table 7-189 Create Rule Configurations

Field Name	Description
Service Name	Specifies the selected PCF service for which a new load shedding rule is to be created.
Name of the rule	Provide a unique name for the load shedding rule to be created. The name can only contain the characters A–Z, a–z, 0–9, period (.), hyphen (-), and underline (_). The maximum length is 255 characters.

   b. Click **Create** to create the load shedding rule.
   To discard the changes, click **Cancel**.
   c. The value gets listed on the Congestion Load Shedding Rules page.
   By default, the newly created rule has the same values as the default rule configuration values.

Figure 7-51 Bulwark Default Load Shedding Rule



d. Click  to view the Discard Priority values for the various congested states. The Bulwark pod's can be in following Congestion states:

Table 7-190 Bulwark Congestion States

Congestion States	Discard Priority
DANGER_OF_CONGESTION (DOC)	20
CONGESTED	10

The Binding service pod's can be in following Congestion states:

Table 7-191 Binding Service Congestion States

Congestion States	Discard Priority
DANGER_OF_CONGESTION (DOC)	30
CONGESTION_L1	27
CONGESTION_L2	24
CONGESTED	20

The SM service pod's can be in following Congestion states:

Table 7-192 SM Policy Service Congestion States

Congestion States	Discard Priority
DANGER_OF_CONGESTION (DOC)	27
CONGESTION_L1	19
CONGESTION_L2	17
CONGESTED	15

The Policy service pod's can be in following Congestion states:

Table 7-193 Policy Service Congestion States

Congestion States	Discard Priority
DANGER_OF_CONGESTION (DOC)	28
CONGESTION_L1	20
CONGESTION_L2	17
CONGESTED	15

e. Click  to edit each of the various congested states **Discard Priority** value. This opens the **Edit <Congestion States>** page, where <Congestion States> will be different for different services.

f. Enter the following information:

Table 7-194 Edit <Congestion State> Fields

Field Name	Description
Service Name	Specifies the selected PCF service for which the load shedding rule is edited.
Rule Name	Specifies the selected load shedding rule for the chosen Congestion state.
Discard Priority	Specifies the discard priority that can be configured for the selected Congestion state. This number can be between 1 to 100 for Bulwark, Binding and PDS services. This number can be between 1 to 31 for Usage Monitoring and SM services.

g. Click **Save** to save the configurations.
To discard the changes, click **Cancel**.

The updated value gets listed on the Congestion Load Shedding Rules page.

5. Click  . This opens the **Copy Rule** window to copy an existing profile.
While using the copy rule option, whichever rule is selected in the **Load Shedding Rules** drop down, the values are copied from that rule.

6. Click  . This deletes the selected load shedding rule. You cannot delete the default system provided rule.

7.10.4 Overload Control Threshold

To open Overload Control Threshold page:

1. From the navigation menu under **Policy**, navigate to **Overload Control Configurations**, and select **Overload Control Threshold**.

   Overload Control Threshold page is displayed. The page shows the existing configurations.

2. Policy allows you to configure the threshold values using a profile. You can either use a default or custom profile or create a new profile. You must activate one of the profiles to use the values. At a time, you can activate only one profile.

> **Note**
>
> If there are no profiles activated, Policy calculates the load level based on `overloadLevelThreshold` configured in `custom-values.yaml` file.

If you are upgrading from an older version of Policy to 23.1.0 or later, the following message appears at the top of the page:

```
System looks to be using old configuration (Not profile based). Request you to
please migrate the data or activate one of the profile
```

a. Click **Migrate**.
Migrate Data window opens.

b. Enter the name of the profile.

c. Either click **Migrate** to migrate the data from the previous version of Policy or click **Migrate and Activate** to migrate the data and activate the profile.

> **Note**
>
> Migrate button will only migrate the existing threshold data to profile. It does not activate the profile. Until one of the profile is not active , the UI will continue to show the above message to migrate the data.
>
> In case of fault recovery, the active profile details are recovered and used from the database. No need to migrate again.

Table 7-195 Create and Manage Threshold Profiles

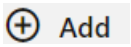
Button/icon	Action
	Opens the **Create Profile** window and allows you to create a new profile. By default, the values of the default profile are associated with the new profile. Click  next to each of the services to edit the values.
	Deletes the selected threshold profile. **Note**: You cannot delete the default system provided profile.
	Opens the **Copy Profile** window to copy an existing profile. While using the copy profile option, whichever profile is selected in the **Threshold Profile** dropdown, the values are copied from that profile.
Threshold Profile	Lists all the threshold profiles. You can select any of the profiles from the list to activate. If the selected profile is already active, you can see **(ACTIVE)** button in green color next to the profile name.
Activate	Activates the profile selected from **Threshold Profile** list. At a time, you can have only one active profile.
Go To Active	**Go To Active** button selects the current active profile from the **Threshold Profile** drop down list. **Note**: If none of the profiles are active, the **Go To Active** button does not appear.

Table 7-196 Configure Threshold Values

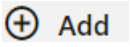
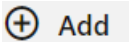
Configuration	Description
Service Name	You can configure the threshold values for the following services: • PCRF Core • Diameter Connector • PCF Session Management • PCF Access and Mobility • PCF UE Policy • CHF Connector • UDR Connector **Note**: Irrespective of PCF or PCRF mode of deployment, this list of services remains the same. For PCRF-CORE deployment mode, configuring overload level threshold for SM Service, AM Service, and UE Policy Service is not required. For PCF deployment mode, configuring the overload level threshold for PCRF Core is not required.
CPU	Click  to view the abatement and onset values for each of the three levels (L1, L2, and L3). The onset and abatement values for CPU are calucalted in percentage (%) and the range is from 1 to 100. Click  to edit the abatement and onset values for each of the levels. **Note**: Make sure that the onset value of L1 is less than the abatement value of L2 and the onset value of L2 is less than the abatement value of L3. You can click  to delete the CPU values for a service. The application prompts you to confirm before deleting the values. **Note**: If the above mentioned CPU validation fails for any of the threshold levels, importing of the Overload Threshold Profile will be rejected.
Pending Message Count	Click  to add the pending message count for each of the services. Pending message count accepts an integer value between 1 to 1000000.
Failure Count	Click  to add the failure count for each of the services. The failure count accepts an integer value between 1 to 1000000.

Table 7-196 (Cont.) Configure Threshold Values

Configuration	Description
Memory	Click Add to add the abatement and onset memory values for each of the three levels (L1, L2, and L3). Memory details are calculated in Percentage (%) and ranges between 1 to 100.

7.11 Controlled Shutdown Configurations

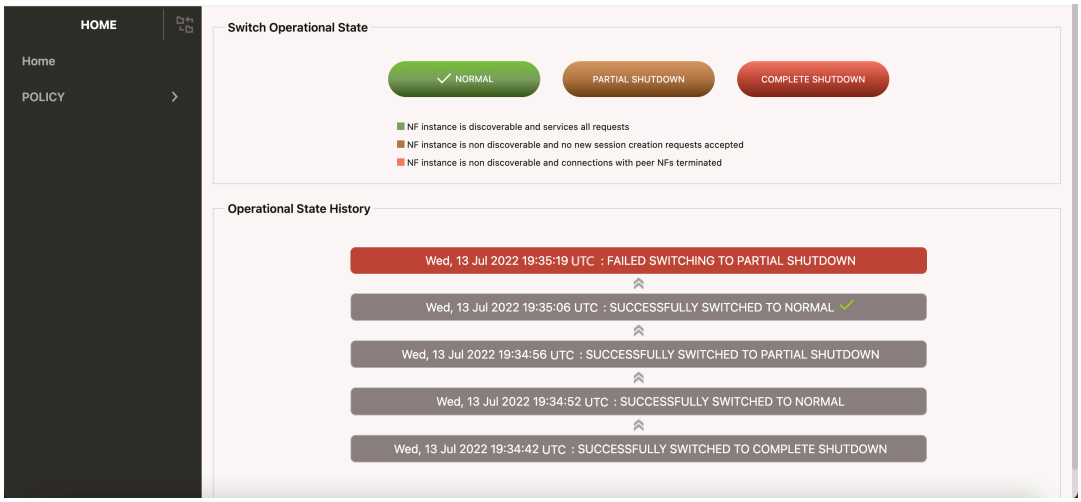
This section describes how to perform Controlled Shutdown configurations for Diameter and Ingress interface.

7.11.1 Operational State

To change Operational State of a site:

- From the navigation menu, under **Policy**, click **Controlled Shutdown**, and then select **Operational State**.
  This opens the page displaying the two groups, **Switch Operational State** and **Operational State History**:

Figure 7-52 Operational State



  - Switch Operational State It displays the following operational states:
    * NORMAL: NF instance is discoverable and services all requests.
    * PARTIAL SHUTDOWN: NF instance is non discoverable and no new session creation requests accepted.
    * COMPLETE SHUTDOWN: NF instance is non discoverable and no new session creation requests accepted.

> **Note**
>
> By default, NORMAL state is assigned to a site. The current state of any site can be identified with a tick mark.

> **Note**
>
> The operational state is stored in configuration server. The service instances (pod) detects the state change only after the configuration refresh is performed. Hence if the config refresh interval is set at 5 seconds then te pods recognise the operational state change after 5 seconds.

You can switch to a different operational state by clicking the NORMAL, PARTIAL SHUTDOWN, or COMPLETE SHUTDOWN button.

– Operational State History: It displays the history of the operational states along with the timestamp.

> **Note**
>
> It displays maximim of ten records at a time. On scrolling further, another set of ten records is displayed. The maximum number of record maintained is hundred.

7.11.2 Diameter Error Mapping

To configure Diameter Error Mapping, perform the following steps:

1. From the navigation menu under **Policy**, click **Controlled Shutdown** and then select **Diameter Error Mapping**.
   This opens the **Diameter Error Mapping** page. The page lists the existing configurations. You can add or import new diameter error mapping configurations using this page.

   > **Note**
   >
   > Click Export to download the available listings in the JSON file format on your system.

2. Click Add .

   This opens the Create Diameter Error Mapping page.

3. On the **Create Diameter Error Mapping** page, enter the following information:

Table 7-197 Create Diameter Error Mapping

Field Name	Description
Message Type	Type of the request

Table 7-197 (Cont.) Create Diameter Error Mapping

Field Name	Description
Answer with Result Code	Specifies the result code, returned in the answer response, when request message is rejected as part of congestion control. Users can select any of the following values from the drop-down menu: • DIAMETER_TOO_BUSY • DIAMETER_UNABLE_TO_COMPLY • DIAMETER_UNABLE_TO_DELIVER • CUSTOM_RESULT_CODE **Note**: When the CUSTOM_RESULT_CODE value is selected, the following two fields are populated on the page: • **Result Code**: Enter a custom result code. • **Use Experimental Result Code**: This is disabled by default. You can enable it by clicking the icon against it. When it is enabled, Vendor ID field is poplulated on the page: – **Vendor ID**: Enter a valid value to specify vendor ID.

4. Click **Save**.
   The configuration gets listed on the **Diameter Error Mapping** page. The page defines the Diameter Error Mapping configuration in the Policy database and it is available to be used in a Policy.

> **Note**
>
> Use  or  available under the **Actions** column to update or delete the configuration.

Importing Diameter Error Mapping

To import Diameter Error Mapping configuration:

1. Click  .
   The page opens the **File Upload** dialog box.
2. Upload the file in JSON format by using the **Drag and Drop** button.
3. Click **Import**.

7.11.3 SBI Ingress Error Mapping

To configure SBI Ingress Error Mapping, perform the following steps:

1. From the navigation menu, under **Policy**, click **Controlled Shutdown**, and then select **SBI Ingress Error Mapping**.
   This opens the SBI Ingress Error Mapping page.
2. Click Edit.
   This opens the Edit SBI Ingress Error Mapping page.

3. Click Add .

   This opens the Add SBI Ingress Error Mapping page.

4. Enter values for the available input fields as described in the following table:

Table 7-198 Ingress Error Mapping Configurations

Field Name	Description
Id	Specifies the list of IDs available for Policy.
Error Code Profile	Select an error code profile from the dropdown list. It displays the list of error profiles configured using the SBI Ingress Error Code Profiles Collection.

5. Click **Save** to save the Ingress error mapping.
   To discard the changes, click **Cancel**.

   The value gets listed on the SBI Ingress Error Mapping page. Use or available under the **Actions** column to update or delete the profile.

7.12 NF Scoring Configurations

You can configure the NF Scoring feature using the CNC Console. To navigate to NF Scoring, click NF Scoring, under Policy. It shows Settings and Calculated Score, which are described as follows:

Settings:

Table 7-199 CNC Console NF Scoring Settings

Field Name	Description
Enable NF Scoring	Specifies whether to enable or disable the NF Scoring
TPS	
Enable	Enables the TPS.
Max Score	Specifies the maximum score of the TPS.
Max TPS	Specifies the maximum TPS.
Service Health	Specifies the service health of a site.
Enable	Enables the Service Health.
Max Score	Specifies the maximum score of the Service Health.
Signaling Connections	Specifies the Signaling Connections of a site.
Enable	Enables the Signaling Connections.
Max Score	Specifies the maximum score of the Signaling Connections.
Max Connections	Specifies the maximum connections.
Replication Health	Specifies the Replication Health of a site.
Enable	Enables the Replication Health.
Max Score	Specifies the maximum score of the Replication Health.
Locality/Site Preference	Specifies the Locality or Site Preference.

Table 7-199 (Cont.) CNC Console NF Scoring Settings

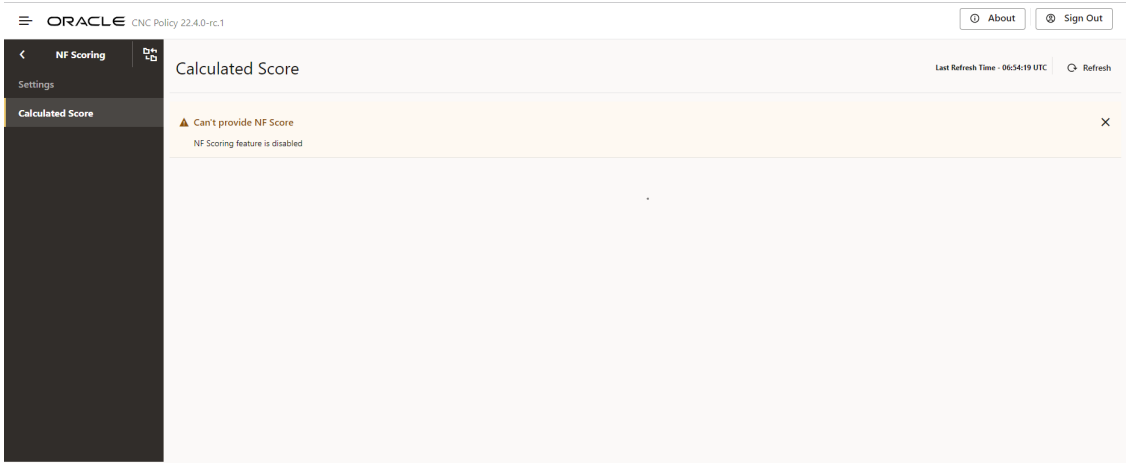
Field Name	Description
Enable	Enables the Locality or Site Preference.
Score	Specifies the score of the Locality or Site Preference.
Active Alert	Specifies the Active Alerts of a site.
Enable	Enables the Active Alert.
Critical Alert Weightage	The site with more critical alerts is unhealthy.
Major Alert Weightage	The site with more major alerts is unhealthy.
Minor Alert Weightage	The site with more minor alerts is unhealthy.

Calculated Score:

To check the calculated Score of a site:

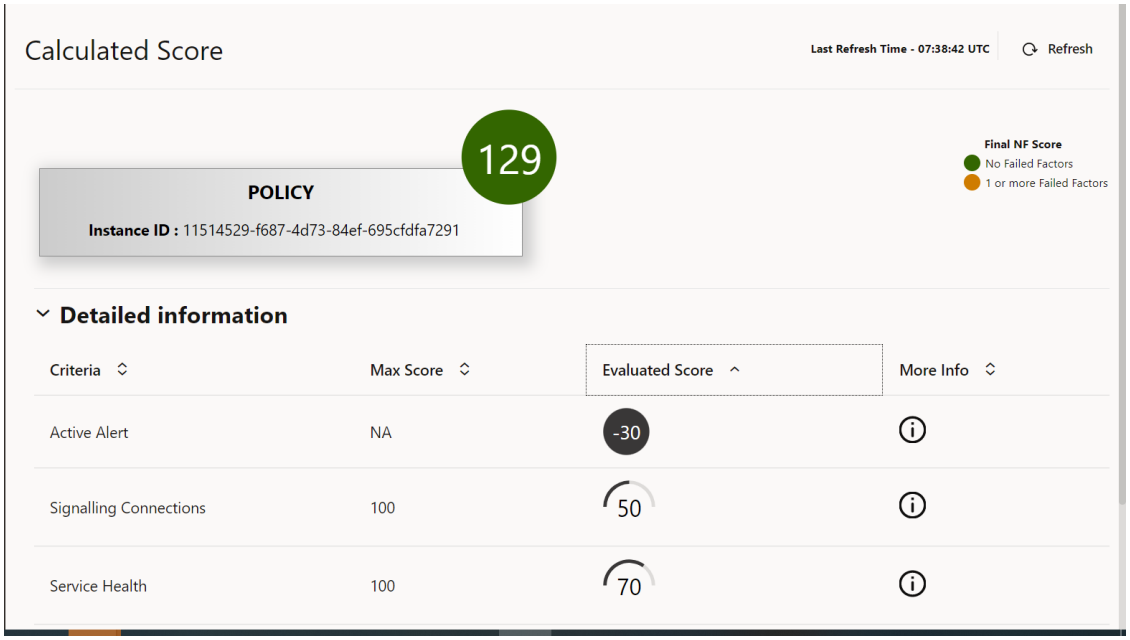
From the navigation menu, under Policy, click NF Scoring and then select Calculated Score. This opens the page displaying the Calculated Score.

Figure 7-53 NF Scoring Disabled



If the NF Scoring feature is enabled. The calculated Screen page displays as:

Figure 7-54 NF Scoring Enabled



> **Note**
>
> If app-info pod is down, you will not be getting the NF Score. You will get an error message that "Data can't be fetched due to internal server error".

Calculated Score shows the total score along with the Instance ID. The total score is shown in either Green or Orange color. If the NF Score is shown in green color there are no failed factors. And, if the NF Score is shown in Orange color there are one or more failed factors. You can click on **Detailed information** to view different criteria and their Max Score, Evaluated Score, and More Info. The criteria in the detailed information tab show the evaluated score. The failed factors are shown with a warning symbol under the evaluated score.

On the top-right of the screen, the *Last Refresh Time* information is available. Moreover, a *Refresh* button is given to refresh the NF Score of a site.

7.13 Viewing cnDBTier Functionalities in CNC Console

Perform the following procedure to view the cnDBTier version, status of cnDBTier clusters, and georeplication status on the CNC Console.

> **Note**
>
> The following cnDBTier functionalities are read only and is available only through CNC Console.

1. From the left navigation pane, click the **Policy** tab, and then click the **DB Tier** tab.
2. Click the **Backup List** to view the list of completed backups along with Backup ID, Backup size, and Creation Timestamp.

The **Backup List** screen is displayed.

Table 7-200 Backup List

Fields	Description
Backup Details	This field displays information such as backup Id, backup size, and backup creation timestamp.
Site Name	This field displays the name of the current site to which Policy is connected.
Backup Id	This field displays the ID of the stored backup.
Backup Size (bytes)	This field displays the size of the stored backup.
Creation TimeStamp	This field displays the time recorded when the backup was stored.

3. Click the **cnDBTier Version** to view the version.

Table 7-201 cnDBTier Version Attributes

Fields	Description
cnDBTier Version	This field displays the cnDBTier version.
NDB Version	This field displays the network database (NDB) version.

4. Click the **Database Statistics Report** to view the available database.

Table 7-202 Database Statistics Report

Fields	Description
Database Count	This field displays the number of available database.
Database Tables Count	This field displays the available database names and their table count.
Database Name	This field displays the database name.
Table Count	This field displays the table count for each database.
Database Table Rows Count	This field displays the table rows present in each table.

a. Click the **View** icon available next to the database name to view the **View Database Table Rows Count** screen.

Table 7-203 View Database Table Rows Count

Fields	Description
Database Name	This field displays the database name.
Tables	This field displays the table names and the corresponding rows in each table.
Table Name	This field displays the table name.
Row Count	This field displays the table rows present in each table.

5. Click the **Georeplication Status** to view the local site and remote site name to which Policy is connected.

Table 7-204 GeoReplication Status

Fields	Description
Local Site Name	This field displays the local site name to which Policy is connected.
Remote Site Name	This field displays the remote site name.
Replication Status	This field displays the replication status with corresponding sites.
Seconds Behind Remote Site	This field displays the number of seconds that the last record read by the local site is behind the latest record written by the remote site for all the replication groups.

a. Click the **View** icon in the **Actions** menu to view the **View Georeplication Status** screen.

Table 7-205 Georeplication Status

Fields	Description
Replication Group Delay	This field displays the number of seconds that the last record read by the local site is behind the latest record written by the remote site for individual replication groups.
Replication Channel Group Id	This field displays the ID of the replication channel group.

b. Click the **View** icon to view the **Replication Group Delay** attributes.

Table 7-206 View Replication Group Delay

Fields	Description
Channel Details	This field displays the channel details such as Remote Replication IP and Role.
Remote Replication IP	This field displays the IP of the remote replication channel.
Role	This field displays the role of the replication channel IP.

6. Click the **HeartBeat Status** to view the connectivity between local site and remote site to which Policy is connected.

Table 7-207 HeartBeat Status Details

Fields	Description
Site Name	This field displays the name of the current site to which Policy is connected.
HeartBeat Details	This field displays information such as the remote site name, heartbeat status, heartbeat lag, and replication channel group id.
Remote Site Name	This field displays the remote site name.
Heartbeat Status	This field displays the connectivity status with corresponding sites.
Heartbeat Lag	This field displays the lag or latency in seconds it took to syncronize between sites.
Replication Channel Group Id	This field displays the ID of the replication channel group.

7. Click the **Local Cluster Status** to view the local cluster status for the current site:

Table 7-208 Local Cluster Status

Fields	Description
Site Name	This field displays the name of the current site to which Policy is connected.
Cluster Status	This field displays the local cluster status for the current site.

8. Click the **On Demand Backup** to create a new backup and view the status of initiated on-demand backups.

> **Note**
>
> On Demand Backup can be initiated on both single site and multi-site cnDBTier cluster and can be used to restore the first standalone site. DB Backup will not be initiated if sites are not properly configured.

Table 7-209 On Demand Backup Details

Fields	Description
Site Name	This field displays the name of the current site to which Policy is connected.
DR Status	This field displays the status of DR.
Backup Id	This field displays the ID of the stored backup.
Backup Status	This field displays the status of backup.
Remote Transfer Status	The field displays the status of remote transfer.
Initiate Backup	The field displays whether the backup is initiated or not.

a. Click the **Edit** icon.
The **Edit** On Demand Backup screen appears.

> **Note**
>
> The **Edit** mode is available only for Initiate Backup.

b. To enable the Initiate Backup option, click **Save**.
A confirmation message "Save successfully" appears.

c. Click **Cancel** to navigate back to the On Demand Backup screen.

d. Click **Refresh** to reload the On Demand Backup screen.

8 Alerts

This section provides information on Policy alerts and their configuration.

> **Note**
>
> The performance and capacity of the system can vary based on the call model, configuration, including but not limited to the deployed policies and corresponding data, for example, policy tables.

8.1 Configuring Alerts

This section describes how to configure alerts in Policy. The Alert Manager uses the Prometheus measurements values as reported by microservices in conditions under alert rules to trigger alerts.

Note

- Sample alert files are packaged with Policy Custom Templates. The `Policy Custom Templates.zip` file can be downloaded from MOS. Unzip the folder to access the following files:
  - Common_Alertrules_cne1.9+.yaml
  - PCF_Alertrules_cne1.9+.yaml
  - PCRF_Alertrules_cne1.9+.yaml
- Name in the metadata section should be unique while applying more than one unique files. For example:

```
apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
  creationTimestamp: null
  labels:
    role: cnc-alerting-rules
  name: occnp-pcf-alerting-rules
```

- If required, edit the threshold values of various alerts in the alert files before configuring the alerts.
- The Alert Manager and Prometheus tools should run in CNE namespace, for example, occne-infra.
- Use the following table to select the appropriate files on the basis of deployment mode and CNE version

Table 8-1 Alert Configuration

Deployment Mode	CNE 1.9+
Converged Mode	Common_Alertrules_cne1.9+.yaml PCF_Alertrules_cne1.9+.yaml PCRF_Alertrules_cne1.9+.yaml
PCF only	Common_Alertrules_cne1.9+.yaml PCF_Alertrules_cne1.9+.yaml
PCRF only	Common_Alertrules_cne1.9+.yaml PCRF_Alertrules_cne1.9+.yaml

Configuring Alerts in Prometheus for CNE 1.9.0 and later versions

To configure PCF alerts in Prometheus for CNE 1.9.0, perform the following steps:

1. Copy the the required file to the Bastion Host.

2. To create or replace the PrometheusRule CRD, run the following command:

```
$ kubectl apply -f Common_Alertrules_cne1.9+.yaml -n <namespace>
```

```
$ kubectl apply -f PCF_Alertrules_cne1.9+.yaml -n <namespace>
```

```
$ kubectl apply -f PCRF_Alertrules_cne1.9+.yaml -n <namespace>
```

> **Note**
>
> This is a sample command for Converged mode of deployment.

To verify if the CRD is created, run the following command:

```
kubectl get prometheusrule -n <namespace>
```

Example:

```
kubectl get prometheusrule -n occnp
```

3. Verify the alerts in the Prometheus GUI. To do so, select the Alerts tab, and view alert details by selecting any individual rule from the list.

Validating Alerts

After configuring the alerts in Prometheus server, a user can verify using the following procedure:

- Open the Prometheus server from your browser using the <IP>:<Port>
- Navigate to Status and then Rules
- Search Policy. Policy Alerts list is displayed.

If you are unable to see the alerts, verify if the alert file is correct and then try again.

Adding worker node name in metrics

To add the worker node name in metrics, perform the following steps:

1. Edit the configmap `occne-prometheus-server` in namespace - `occne-infra`.
2. Locate the the following job:

```
job_name: kubernetes-pods
kubernetes_sd_configs:
role: pod
```

3. Add the following in the `relabel_configs`:

```
action: replace
source_labels:
__meta_kubernetes_pod_node_name
target_label: kubernetes_pod_node_name
```

8.2 Configuring SNMP Notifier

This section describes the procedure to configure SNMP Notifier.

Configure the IP and port of the SNMP trap receiver in the SNMP Notifier using the following procedure:

1. Run the following command to edit the deployment:

   ```
   $ kubectl edit deploy <snmp_notifier_deployment_name> -n <namespace>
   ```

   Example:

   ```
   $ kubectl edit deploy occne-snmp-notifier -n occne-infra
   ```

   SNMP deployment yaml file is displayed.
2. Edit the SNMP destination in the deployment yaml file as follows:

   ```
   --snmp.destination=<destination_ip>:<destination_port>
   ```

   Example:

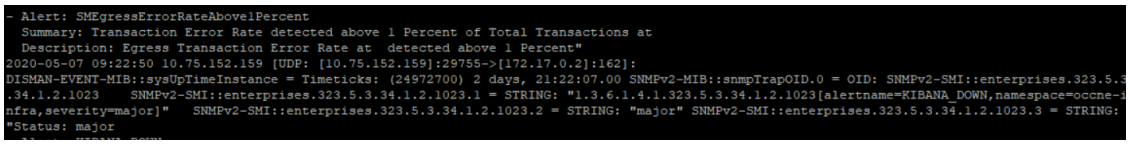
   ```
   --snmp.destination=10.75.203.94:162
   ```
3. Save the file.

Checking SNMP Traps
Following is an example on how to capture the logs of the trap receiver server to view the generated SNMP traps:

```
$ docker logs <trapd_container_id>
```

Sample output:

Figure 8-1 Sample output for SNMP Trap



MIB Files for Policy
There are two MIB files which are used to generate the traps. Update these files along with the Alert file in order to fetch the traps in their environment.

- `toplevel.mib`
  This is the top level mib file, where the Objects and their data types are defined.
- `policy-alarm-mib.mib`
  This file fetches objects from the top level mib file and these objects can be selected for display.

> **Note**
>
> MIB files are packaged along with Custom Templates. Download the file from MOS. For more information on downloading custom templates, see *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.

8.3 List of Alerts

This section provides detailed information about the alert rules defined for Policy. It consists of the following three types of alerts:

1. Common Alerts - This category of alerts is common and required for all three modes of deployment.
2. PCF Alerts - This category of alerts is specific to PCF microservices and required for Converged and PCF only modes of deployment.
3. PCRF Alerts - This category of alerts is specific to PCRF microservices and required for Converged and PCRF only modes of deployment.

8.3.1 Common Alerts

This section provides information about alerts that are common for PCF and PCRF.

8.3.1.1 POD_CONGESTION_L1

Table 8-2 POD_CONGESTION_L1

Field	Details
Name in Alert Yaml File	PodCongestionL1
Description	Alert when cpu of pod is in CONGESTION_L1 state.
Summary	Alert when cpu of pod is in CONGESTION_L1 state.
Severity	Critical
Condition	occnp_pod_resource_congestion_state{type="cpu",container!~"bulwark|diam-gateway"} == 2
OID	1.3.6.1.4.1.323.5.3.52.1.2.71
Metric Used	occnp_pod_resource_congestion_state
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.2 POD_CONGESTION_L2

Table 8-3 POD_CONGESTION_L2

Field	Details
Name in Alert Yaml File	PodCongestionL2
Description	Alert when cpu of pod is in CONGESTION_L2 state.
Summary	Alert when cpu of pod is in CONGESTION_L2 state.
Severity	Critical

Table 8-3 (Cont.) POD_CONGESTION_L2

Field	Details
Condition	occnp_pod_resource_congestion_state{type="cpu"} == 3
OID	1.3.6.1.4.1.323.5.3.52.1.2.72
Metric Used	occnp_pod_resource_congestion_state
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.3 POD_PENDING_REQUEST_CONGESTION_L1

Table 8-4 POD_PENDING_REQUEST_CONGESTION_L1

Field	Details
Name in Alert Yaml File	PodPendingRequestCongestionL1
Description	Alert when queue of pod is in CONGESTION_L1 state.
Summary	Alert when queue of pod is in CONGESTION_L1 state.
Severity	critical
Condition	occnp_pod_resource_congestion_state{type="queue",container!~"bulwark|diam-gateway"} == 2
OID	1.3.6.1.4.1.323.5.3.52.1.2.73
Metric Used	occnp_pod_resource_congestion_state
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.4 POD_PENDING_REQUEST_CONGESTION_L2

Table 8-5 POD_PENDING_REQUEST_CONGESTION_L2

Field	Details
Name in Alert Yaml File	PodPendingRequestCongestionL2
Description	Alert when queue of pod is in CONGESTION_L2 state.
Summary	Alert when queue of pod is in CONGESTION_L2 state.
Severity	critical
Condition	occnp_pod_resource_congestion_state{type="queue"} == 3
OID	1.3.6.1.4.1.323.5.3.52.1.2.74
Metric Used	occnp_pod_resource_congestion_state
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.5 POD_CPU_CONGESTION_L1

Table 8-6 POD_CPU_CONGESTION_L1

Field	Details
Name in Alert Yaml File	PodCPUCongestionL1
Description	Alert when cpu of pod is in CONGESTION_L1 state.

Table 8-6 (Cont.) POD_CPU_CONGESTION_L1

Field	Details
Summary	Alert when cpu of pod is in CONGESTION_L1 state.Alert when pod is in CONGESTION_L1 state.
Severity	Critical
Condition	occnp_pod_resource_congestion_state{type="cpu",container!~"bulwark|diam-gateway"} == 2
OID	1.3.6.1.4.1.323.5.3.52.1.2.73
Metric Used	occnp_pod_resource_congestion_state
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.6 POD_CPU_CONGESTION_L2

Table 8-7 POD_CPU_CONGESTION_L2

Field	Details
Name in Alert Yaml File	PodCPUCongestionL2
Description	Alert when cpu of pod is in CONGESTION_L2 state.
Summary	Alert when cpu of pod is in CONGESTION_L2 state.
Severity	critical
Condition	occnp_pod_resource_congestion_state{type="cpu"} == 3
OID	1.3.6.1.4.1.323.5.3.52.1.2.74
Metric Used	occnp_pod_resource_congestion_state
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.7 PodMemoryDoC

Table 8-8 PodMemoryDoC

Field	Details
Description	Pod Resource Congestion status of {{$labels.service}} service is DoC for Memory type
Summary	Pod Resource Congestion status of {{$labels.service}} service is DoC for Memory type
Severity	Major
Condition	occnp_pod_resource_congestion_state{type="memory"} == 1
OID	1.3.6.1.4.1.323.5.3.52.1.2.31
Metric Used	occnp_pod_resource_congestion_state

Table 8-8 (Cont.) PodMemoryDoC

Field	Details
Recommended Actions	Alert triggers based on the resource limit usage and load shedding configurations in congestion control. The CPU, Memory, and queue usage can be referred using the Grafana Dashboard. **Note** Threshold levels can be configured using the `PCF_Alertrules.yaml` file. For any additional guidance, contact My Oracle Support.

8.3.1.8 PodMemoryCongested

Table 8-9 PodMemoryCongested

Field	Details
Description	Pod Resource Congestion status of {{$labels.service}} service is congested for Memory type
Summary	Pod Resource Congestion status of {{$labels.service}} service is congested for Memory type
Severity	Critical
Condition	occnp_pod_resource_congestion_state{type="memory"} == 2
OID	1.3.6.1.4.1.323.5.3.52.1.2.32
Metric Used	occnp_pod_resource_congestion_state
Recommended Actions	Alert triggers based on the resource limit usage and load shedding configurations in congestion control. The CPU, Memory, and queue usage can be referred using the Grafana Dashboard. For any additional guidance, contact My Oracle Support.

8.3.1.9 PodDoc

Table 8-10 PodDoc

Field	Details
Description	Pod Congestion status of {{$labels.service}} service is DoC.
Summary	Pod Congestion status of {{$labels.service}} service is DoC.
Severity	Major
Condition	occnp_pod_congestion_state == 1
OID	1.3.6.1.4.1.323.5.3.52.1.2.25
Metric Used	occnp_pod_congestion_state
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.10 RAA_RX_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Table 8-11 RAA_RX_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Field	Details
Description	RAA Rx fail count exceeds the critical threshold limit.
Summary	RAA Rx fail count exceeds the critical threshold limit.
Severity	CRITICAL
Condition	sum(rate(occnp_diam_response_local_total{msgType="RAA", appId="16777236", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="RAA", appId="16777236"}[5m])) * 100 > 90
OID	1.3.6.1.4.1.323.5.3.52.1.2.35
Metric Used	occnp_diam_response_local_total
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.11 RAA_RX_FAIL_COUNT_EXCEEDS_MAJOR_THRESHOLD

Table 8-12 RAA_RX_FAIL_COUNT_EXCEEDS_MAJOR_THRESHOLD

Field	Details
Description	RAA Rx fail count exceeds the major threshold limit.
Summary	RAA Rx fail count exceeds the major threshold limit.
Severity	MAJOR
Condition	sum(rate(occnp_diam_response_local_total{msgType="RAA", appId="16777236", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="RAA", appId="16777236"}[5m])) * 100 > 80 and sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="RAA",responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="RAA"}[5m])) * 100 <= 90
OID	1.3.6.1.4.1.323.5.3.52.1.2.35
Metric Used	occnp_diam_response_local_total
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.12 RAA_RX_FAIL_COUNT_EXCEEDS_MINOR_THRESHOLD

Table 8-13 RAA_RX_FAIL_COUNT_EXCEEDS_MINOR_THRESHOLD

Field	Details
Description	RAA Rx fail count exceeds the minor threshold limit.
Summary	RAA Rx fail count exceeds the minor threshold limit.
Severity	MINOR

Table 8-13 (Cont.) RAA_RX_FAIL_COUNT_EXCEEDS_MINOR_THRESHOLD

Field	Details
Condition	sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="RAA",responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="RAA"}[5m])) * 100 > 60 and sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="RAA",responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="RAA"}[5m])) * 100 <= 80
OID	1.3.6.1.4.1.323.5.3.52.1.2.35
Metric Used	occnp_diam_response_local_total
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.13 ASA_RX_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Table 8-14 ASA_RX_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Field	Details
Description	ASA Rx fail count exceeds the critical threshold limit.
Summary	ASA Rx fail count exceeds the critical threshold limit.
Severity	CRITICAL
Condition	sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA",responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA"}[5m])) * 100 > 90
OID	1.3.6.1.4.1.323.5.3.52.1.2.66
Metric Used	occnp_diam_response_local_total
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.14 ASA_RX_FAIL_COUNT_EXCEEDS_MAJOR_THRESHOLD

Table 8-15 ASA_RX_FAIL_COUNT_EXCEEDS_MAJOR_THRESHOLD

Field	Details
Description	ASA Rx fail count exceeds the major threshold limit.
Summary	ASA Rx fail count exceeds the major threshold limit.
Severity	MAJOR
Condition	sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA",responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA"}[5m])) * 100 > 80 and sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA",responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA"}[5m])) * 100 <= 90
OID	1.3.6.1.4.1.323.5.3.52.1.2.66
Metric Used	occnp_diam_response_local_total

Table 8-15 (Cont.) ASA_RX_FAIL_COUNT_EXCEEDS_MAJOR_THRESHOLD

Field	Details
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.15 ASA_RX_FAIL_COUNT_EXCEEDS_MINOR_THRESHOLD

Table 8-16 ASA_RX_FAIL_COUNT_EXCEEDS_MINOR_THRESHOLD

Field	Details
Description	ASA Rx fail count exceeds the minor threshold limit.
Summary	ASA Rx fail count exceeds the minor threshold limit.
Severity	MINOR
Condition	sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA",responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA"}[5m])) * 100 > 60 and sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA",responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA"}[5m])) * 100 <= 80
OID	1.3.6.1.4.1.323.5.3.52.1.2.66
Metric Used	occnp_diam_response_local_total
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.16 ASA_RX_TIMEOUT_COUNT_EXCEEDS_MINOR_THRESHOLD

Table 8-17 ASA_RX_TIMEOUT_COUNT_EXCEEDS_MINOR_THRESHOLD

Field	Details
Description	ASA Rx timeout count exceeds the minor threshold limit
Summary	ASA Rx timeout count exceeds the minor threshold limit
Severity	MINOR
Condition	sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA",responseCode="timeout"}[5m])) / sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA"}[5m])) * 100 > 60 and sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA",responseCode="timeout"}[5m])) / sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA"}[5m])) * 100 <= 80
OID	1.3.6.1.4.1.323.5.3.52.1.2.67
Metric Used	
Recommended Actions	

8.3.1.17 ASA_RX_TIMEOUT_COUNT_EXCEEDS_MAJOR_THRESHOLD

Table 8-18 ASA_RX_TIMEOUT_COUNT_EXCEEDS_MAJOR_THRESHOLD

Field	Details
Description	ASA Rx timeout count exceeds the major threshold limit
Summary	ASA Rx timeout count exceeds the major threshold limit
Severity	sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA",responseCode="timeout"}[5m])) / sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA"}[5m])) * 100 > 80 and sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA",responseCode="timeout"}[5m])) / sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA"}[5m])) * 100 <= 90
Condition	MAJOR
OID	1.3.6.1.4.1.323.5.3.52.1.2.67
Metric Used	
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.18 ASA_RX_TIMEOUT_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Table 8-19 ASA_RX_TIMEOUT_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Field	Details
Description	ASA Rx timeout count exceeds the critical threshold limit
Summary	ASA Rx timeout count exceeds the critical threshold limit
Severity	CRITICAL
Condition	sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA",responseCode="timeout"}[5m])) / sum(rate(occnp_diam_response_local_total{appId="16777236",msgType="ASA"}[5m])) * 100 > 90
OID	1.3.6.1.4.1.323.5.3.52.1.2.67
Metric Used	
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.19 SCP_PEER_UNAVAILABLE

Table 8-20 SCP_PEER_UNAVAILABLE

Field	Details
Description	Configured SCP peer is unavailable.
Summary	Configured SCP peer is unavailable.
Severity	Major
Condition	occnp_oc_egressgateway_peer_health_status != 0. SCP peer [{{$labels.peer}}] is unavailable.
OID	1.3.6.1.4.1.323.5.3.52.1.2.60
Metric Used	occnp_oc_egressgateway_peer_health_status

Table 8-20 (Cont.) SCP_PEER_UNAVAILABLE

Field	Details
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.20 SCP_PEER_SET_UNAVAILABLE

Table 8-21 SCP_PEER_SET_UNAVAILABLE

Field	Details
Description	None of the SCP peer available for configured peerset.
Summary	None of the SCP peer available for configured peerset.
Severity	Critical
Condition	One of the SCPs has been marked unhealthy.
OID	1.3.6.1.4.1.323.5.3.52.1.2.61
Metric Used	oc_egressgateway_peer_count and oc_egressgateway_peer_available_count
Recommended Actions	NF clears the critical alarm when atleast one SCP peer in a peerset becomes available such that all other SCP peers in the given peerset are still unavailable. For any additional guidance, contact My Oracle Support.

8.3.1.21 STALE_CONFIGURATION

Table 8-22 STALE_CONFIGURATION

Field	Details
Description	In last 10 minutes, the current service config_level does not match the config_level from the config-server.
Summary	In last 10 minutes, the current service config_level does not match the config_level from the config-server.
Severity	Major
Condition	(sum by(namespace) (topic_version{app_kubernetes_io_name="config-server",topicName="config.level"})) / (count by(namespace) (topic_version{app_kubernetes_io_name="config-server",topicName="config.level"})) != (sum by(namespace) (topic_version{app_kubernetes_io_name!="config-server",topicName="config.level"})) / (count by(namespace) (topic_version{app_kubernetes_io_name!="config-server",topicName="config.level"}))
OID	1.3.6.1.4.1.323.5.3.52.1.2.62
Metric Used	topic_version
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.22 POLICY_SERVICES_DOWN

Table 8-23 POLICY_SERVICES_DOWN

Field	Details
Name in Alert Yaml File	PCF_SERVICES_DOWN
Description	{{$labels.service}} service is not running.

Table 8-23 (Cont.) POLICY_SERVICES_DOWN

Field	Details
Summary	{{$labels.service}} service is not running.
Severity	Critical
Condition	None of the pods of the CNC Policy application are available.
OID	1.3.6.1.4.1.323.5.3.36.1.2.1
Metric Used	appinfo_service_running{vendor="Oracle", application="occnp", category!=""}!= 1
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.23 DIAM_TRAFFIC_RATE_ABOVE_THRESHOLD

Table 8-24 DIAM_TRAFFIC_RATE_ABOVE_THRESHOLD

Field	Details
Name in Alert Yaml File	DiamTrafficRateAboveThreshold
Description	Diameter Connector Ingress traffic Rate is above threshold of Max MPS (current value is: {{ $value }})
Summary	Traffic Rate is above 90 Percent of Max requests per second.
Severity	Major
Condition	The total Ingress traffic rate for Diameter connector has crossed the configured threshold of 900 TPS. Default value of this alert trigger point in Common_Alertrules.yaml file is when Diameter Connector Ingress Rate crosses 90% of maximum ingress requests per second.
OID	1.3.6.1.4.1.323.5.3.36.1.2.6
Metric Used	ocpm_ingress_request_total
Recommended Actions	The alert gets cleared when the Ingress traffic rate falls below the threshold. **Note**: Threshold levels can be configured using the `Common_Alertrules.yaml` file. It is recommended to assess the reason for additional traffic. Perform the following steps to analyze the cause of increased traffic: 1. Refer **Ingress Gateway** section in Grafana to determine increase in 4xx and 5xx error response codes. 2. Check **Ingress Gateway** logs on Kibana to determine the reason for the errors. For any additional guidance, contact My Oracle Support.

8.3.1.24 DIAM_INGRESS_ERROR_RATE_ABOVE_10_PERCENT

Table 8-25 DIAM_INGRESS_ERROR_RATE_ABOVE_10_PERCENT

Field	Details
Name in Alert Yaml File	DiamIngressErrorRateAbove10Percent
Description	Transaction Error Rate detected above 10 Percent of Total on Diameter Connector (current value is: {{ $value }})
Summary	Transaction Error Rate detected above 10 Percent of Total Transactions.
Severity	Critical

Table 8-25 (Cont.) DIAM_INGRESS_ERROR_RATE_ABOVE_10_PERCENT

Field	Details
Condition	The number of failed transactions is above 10 percent of the total transactions on Diameter Connector.
OID	1.3.6.1.4.1.323.5.3.36.1.2.7
Metric Used	ocpm_ingress_response_total
Recommended Actions	The alert gets cleared when the number of failed transactions are below 10% of the total transactions. To assess the reason for failed transactions, perform the following steps: 1. Check the service specific metrics to understand the service specific errors. For instance: `ocpm_ingress_response_total{servicename_3gpp="rx",response_code!~"2.*"}` 2. The service specific errors can be further filtered for errors specific to a method such as GET, PUT, POST, DELETE, and PATCH. For any additional guidance, contact My Oracle Support.

8.3.1.25 DIAM_EGRESS_ERROR_RATE_ABOVE_1_PERCENT

Table 8-26 DIAM_EGRESS_ERROR_RATE_ABOVE_1_PERCENT

Field	Details
Name in Alert Yaml File	DiamEgressErrorRateAbove1Percent
Description	Egress Transaction Error Rate detected above 1 Percent of Total on Diameter Connector (current value is: {{ $value }})
Summary	Transaction Error Rate detected above 1 Percent of Total Transactions
Severity	Minor
Condition	The number of failed transactions is above 1 percent of the total Egress Gateway transactions on Diameter Connector.
OID	1.3.6.1.4.1.323.5.3.36.1.2.8
Metric Used	ocpm_egress_response_total
Recommended Actions	The alert gets cleared when the number of failed transactions are below 1% of the total transactions. To assess the reason for failed transactions, perform the following steps: 1. Check the service specific metrics to understand the errors. For instance: `ocpm_egress_response_total{servicename_3gpp="rx",response_code!~"2.*"}` 2. The service specific errors can be further filtered for errors specific to a method such as GET, PUT, POST, DELETE, and PATCH. For any additional guidance, contact My Oracle Support.

8.3.1.26 UDR_INGRESS_TRAFFIC_RATE_ABOVE_THRESHOLD

Table 8-27 UDR_INGRESS_TRAFFIC_RATE_ABOVE_THRESHOLD

Field	Details
Description	User service Ingress traffic Rate from UDR is above threshold of Max MPS (current value is: {{ $value }})
Summary	Traffic Rate is above 90 Percent of Max requests per second
Severity	Major
Condition	The total User Service Ingress traffic rate from UDR has crossed the configured threshold of 900 TPS. Default value of this alert trigger point in Common_Alertrules.yaml file is when user service Ingress Rate from UDR crosses 90% of maximum ingress requests per second.
OID	1.3.6.1.4.1.323.5.3.36.1.2.9
Metric Used	ocpm_userservice_inbound_count_total{service_resource="udr-service"}
Recommended Actions	The alert gets cleared when the Ingress traffic rate falls below the threshold. **Note**: Threshold levels can be configured using the `Common_Alertrules.yaml` file. It is recommended to assess the reason for additional traffic. Perform the following steps to analyze the cause of increased traffic: 1. Refer **Ingress Gateway** section in Grafana to determine increase in 4xx and 5xx error response codes. 2. Check **Ingress Gateway** logs on Kibana to determine the reason for the errors. For any additional guidance, contact My Oracle Support.

8.3.1.27 UDR_EGRESS_ERROR_RATE_ABOVE_10_PERCENT

Table 8-28 UDR_EGRESS_ERROR_RATE_ABOVE_10_PERCENT

Field	Details
Description	Egress Transaction Error Rate detected above 10 Percent of Total on User service (current value is: {{ $value }})
Summary	Transaction Error Rate detected above 10 Percent of Total Transactions
Severity	Critical
Condition	The number of failed transactions from UDR is more than 10 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.36.1.2.10
Metric Used	ocpm_udr_tracking_response_total{servicename_3gpp="nudr-dr",response_code!~"2.*"}

Table 8-28 (Cont.) UDR_EGRESS_ERROR_RATE_ABOVE_10_PERCENT

Field	Details
Recommended Actions	The alert gets cleared when the number of failure transactions falls below the configured threshold. **Note**: Threshold levels can be configured using the `Common_Alertrules.yaml` file. It is recommended to assess the reason for failed transactions. Perform the following steps to analyze the cause of increased traffic: 1. Refer **Egress Gateway** section in Grafana to determine increase in 4xx and 5xx error response codes. 2. Check **Egress Gateway** logs on Kibana to determine the reason for the errors. For any additional guidance, contact My Oracle Support.

8.3.1.28 POLICYDS_INGRESS_TRAFFIC_RATE_ABOVE_THRESHOLD

Table 8-29 POLICYDS_INGRESS_TRAFFIC_RATE_ABOVE_THRESHOLD

Field	Details
Description	Ingress Traffic Rate is above threshold of Max MPS (current value is: {{ $value }})
Summary	Traffic Rate is above 90 Percent of Max requests per second
Severity	Critical
Condition	The total PolicyDS Ingress message rate has crossed the configured threshold of 900 TPS. 90% of maximum Ingress request rate. Default value of this alert trigger point in Common_Alertrules.yaml file is when PolicyDS Ingress Rate crosses 90% of maximum ingress requests per second.
OID	1.3.6.1.4.1.323.5.3.36.1.2.13
Metric Used	client_request_total **Note**: This is a Kubernetes metric used for instance availability monitoring. If the metric is not available, use similar metrics exposed by the monitoring system.
Recommended Actions	The alert gets cleared when the Ingress traffic rate falls below the threshold. **Note**: Threshold levels can be configured using the `Common_Alertrules.yaml` file. It is recommended to assess the reason for additional traffic. Perform the following steps to analyze the cause of increased traffic: 1. Refer **Ingress Gateway** section in Grafana to determine increase in 4xx and 5xx error response codes. 2. Check **Ingress Gateway** logs on Kibana to determine the reason for the errors. For any additional guidance, contact My Oracle Support.

8.3.1.29 POLICYDS_INGRESS_ERROR_RATE_ABOVE_10_PERCENT

Table 8-30 POLICYDS_INGRESS_ERROR_RATE_ABOVE_10_PERCENT

Field	Details
Description	Ingress Transaction Error Rate detected above 10 Percent of Totat on PolicyDS service (current value is: {{ $value }})

Table 8-30 (Cont.) POLICYDS_INGRESS_ERROR_RATE_ABOVE_10_PERCENT

Field	Details
Summary	Transaction Error Rate detected above 10 Percent of Total Transactions
Severity	Critical
Condition	The number of failed transactions is above 10 percent of the total transactions for PolicyDS service.
OID	1.3.6.1.4.1.323.5.3.36.1.2.14
Metric Used	client_response_total
Recommended Actions	The alert gets cleared when the number of failed transactions are below 10% of the total transactions. To assess the reason for failed transactions, perform the following steps: 1. Check the service specific metrics to understand the service specific errors. For instance: `client_response_total{response!~"2.*"}` 2. The service specific errors can be further filtered for errors specific to a method such as GET, PUT, POST, DELETE, and PATCH. For any additional guidance, contact My Oracle Support.

8.3.1.30 POLICYDS_EGRESS_ERROR_RATE_ABOVE_1_PERCENT

Table 8-31 POLICYDS_EGRESS_ERROR_RATE_ABOVE_1_PERCENT

Field	Details
Description	Egress Transaction Error Rate detected above 1 Percent of Total on PolicyDS service (current value is: {{ $value }})
Summary	Transaction Error Rate detected above 1 Percent of Total Transactions
Severity	Minor
Condition	The number of failed transactions is above 1 percent of the total transactions for PolicyDS service.
OID	1.3.6.1.4.1.323.5.3.36.1.2.15
Metric Used	server_response_total
Recommended Actions	The alert gets cleared when the number of failed transactions are below 10% of the total transactions. To assess the reason for failed transactions, perform the following steps: 1. Check the service specific metrics to understand the service specific errors. For instance: `server_response_total{response!~"2.*"}` 2. The service specific errors can be further filtered for errors specific to a method such as GET, PUT, POST, DELETE, and PATCH. For any additional guidance, contact My Oracle Support.

8.3.1.31 UDR_INGRESS_TIMEOUT_ERROR_ABOVE_MAJOR_THRESHOLD

Table 8-32 UDR_INGRESS_TIMEOUT_ERROR_ABOVE_MAJOR_THRESHOLD

Field	Details
Description	Ingress Timeout Error Rate detected above 10 Percent of Totat towards UDR service (current value is: {{ $value }})

Table 8-32 (Cont.) UDR_INGRESS_TIMEOUT_ERROR_ABOVE_MAJOR_THRESHOLD

Field	Details
Summary	Timeout Error Rate detected above 10 Percent of Total Transactions
Severity	Major
Condition	The number of failed transactions due to timeout is above 10 percent of the total transactions for UDR service.
OID	1.3.6.1.4.1.323.5.3.36.1.2.16
Metric Used	ocpm_udr_tracking_request_timeout_total{servicename_3gpp="nudr-dr"}
Recommended Actions	The alert gets cleared when the number of failed transactions due to timeout are below 10% of the total transactions. To assess the reason for failed transactions, perform the following steps: 1. Check the service specific metrics to understand the service specific errors. For instance: `ocpm_udr_tracking_request_timeout_total{servicename_3gpp="nudr-dr"}` 2. The service specific errors can be further filtered for errors specific to a method such as GET, PUT, POST, DELETE, and PATCH. For any additional guidance, contact My Oracle Support.

8.3.1.32 DB_TIER_DOWN_ALERT

Table 8-33 DB_TIER_DOWN_ALERT

Field	Details
Description	DB cannot be reachable.
Summary	DB cannot be reachable.
Severity	Critical
Condition	Database is not available.
OID	1.3.6.1.4.1.323.5.3.36.1.2.18
Metric Used	appinfo_category_running{category="database"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.33 CPUUsagePerServiceAboveMinorThreshold

Table 8-34 CPUUsagePerServiceAboveMinorThreshold

Field	Details
Description	CPU usage for {{$labels.service}} service is above 60
Summary	CPU usage for {{$labels.service}} service is above 60
Severity	Minor
Condition	A service pod has reached the configured minor threshold (60%) of its CPU usage limits.
OID	1.3.6.1.4.1.323.5.3.36.1.2.19
Metric Used	container_cpu_usage_seconds_total **Note**: This is a Kubernetes used for instance availability monitoring. If the metric is not available, use similar metrics exposed by the monitoring system.

Table 8-34 (Cont.) CPUUsagePerServiceAboveMinorThreshold

Field	Details
Recommended Actions	The alert gets cleared when the CPU utilization falls below the minor threshold or crosses the major threshold, in which case CPUUsagePerServiceAboveMajorThreshold alert shall be raised. **Note**: Threshold levels can be configured using the `PCF_Alertrules.yaml` file. For any additional guidance, contact My Oracle Support.

8.3.1.34 CPUUsagePerServiceAboveMajorThreshold

Table 8-35 CPUUsagePerServiceAboveMajorThreshold

Field	Details
Description	CPU usage for {{$labels.service}} service is above 80
Summary	CPU usage for {{$labels.service}} service is above 80
Severity	Major
Condition	A service pod has reached the configured major threshold (80%) of its CPU usage limits.
OID	1.3.6.1.4.1.323.5.3.36.1.2.20
Metric Used	container_cpu_usage_seconds_total **Note**: This is a Kubernetes used for instance availability monitoring. If the metric is not available, use similar metrics exposed by the monitoring system.
Recommended Actions	The alert gets cleared when the CPU utilization falls below the major threshold or crosses the critical threshold, in which case CPUUsagePerServiceAboveCriticalThreshold alert shall be raised. **Note**: Threshold levels can be configured using the `PCF_Alertrules.yaml` file. For any additional guidance, contact My Oracle Support.

8.3.1.35 CPUUsagePerServiceAboveCriticalThreshold

Table 8-36 CPUUsagePerServiceAboveCriticalThreshold

Field	Details
Description	CPU usage for {{$labels.service}} service is above 90
Summary	CPU usage for {{$labels.service}} service is above 90
Severity	Critical
Condition	A service pod has reached the configured critical threshold (90%) of its CPU usage limits.
OID	1.3.6.1.4.1.323.5.3.36.1.2.21
Metric Used	container_cpu_usage_seconds_total **Note**: This is a Kubernetes used for instance availability monitoring. If the metric is not available, use similar metrics exposed by the monitoring system.
Recommended Actions	The alert gets cleared when the CPU utilization falls below the critical threshold. **Note**: Threshold levels can be configured using the `PCF_Alertrules.yaml` file. For any additional guidance, contact My Oracle Support.

8.3.1.36 MemoryUsagePerServiceAboveMinorThreshold

Table 8-37 MemoryUsagePerServiceAboveMinorThreshold

Field	Details
Description	Memory usage for {{$labels.service}} service is above 60
Summary	Memory usage for {{$labels.service}} service is above 60
Severity	Minor
Condition	A service pod has reached the configured minor threshold (60%) of its memory usage limits.
OID	1.3.6.1.4.1.323.5.3.36.1.2.22
Metric Used	container_memory_usage_bytes **Note**: This is a Kubernetes used for instance availability monitoring. If the metric is not available, use similar metrics exposed by the monitoring system.
Recommended Actions	The alert gets cleared when the memory utilization falls below the minor threshold or crosses the critical threshold, in which case MemoryUsagePerServiceAboveMajorThreshold alert shall be raised. **Note**: Threshold levels can be configured using the `PCF_Alertrules.yaml` file. For any additional guidance, contact My Oracle Support.

8.3.1.37 MemoryUsagePerServiceAboveMajorThreshold

Table 8-38 MemoryUsagePerServiceAboveMajorThreshold

Field	Details
Description	Memory usage for {{$labels.service}} service is above 80
Summary	Memory usage for {{$labels.service}} service is above 80
Severity	Major
Condition	A service pod has reached the configured major threshold (80%) of its memory usage limits.
OID	1.3.6.1.4.1.323.5.3.36.1.2.23
Metric Used	container_memory_usage_bytes **Note**: This is a Kubernetes used for instance availability monitoring. If the metric is not available, use similar metrics exposed by the monitoring system.
Recommended Actions	The alert gets cleared when the memory utilization falls below the major threshold or crosses the critical threshold, in which case MemoryUsagePerServiceAboveCriticalThreshold alert shall be raised. **Note**: Threshold levels can be configured using the `PCF_Alertrules.yaml` file. For any additional guidance, contact My Oracle Support.

8.3.1.38 MemoryUsagePerServiceAboveCriticalThreshold

Table 8-39 MemoryUsagePerServiceAboveCriticalThreshold

Field	Details
Description	Memory usage for {{$labels.service}} service is above 90
Summary	Memory usage for {{$labels.service}} service is above 90

Table 8-39 (Cont.) MemoryUsagePerServiceAboveCriticalThreshold

Field	Details
Severity	Critical
Condition	A service pod has reached the configured critical threshold (90%) of its memory usage limits.
OID	1.3.6.1.4.1.323.5.3.36.1.2.24
Metric Used	container_memory_usage_bytes **Note**: This is a Kubernetes used for instance availability monitoring. If the metric is not available, use similar metrics exposed by the monitoring system.
Recommended Actions	The alert gets cleared when the memory utilization falls below the critical threshold. **Note**: Threshold levels can be configured using the `PCF_Alertrules.yaml` file. For any additional guidance, contact My Oracle Support.

8.3.1.39 POD_CONGESTED

Table 8-40 POD_CONGESTED

Field	Details
Description	Pod Congestion status of {{$labels.service}} service is congested
Summary	Pod Congestion status of {{$labels.service}} service is congested
Severity	Critical
Condition	The pod congestion status is set to congested.
OID	1.3.6.1.4.1.323.5.3.36.1.2.26
Metric Used	occnp_pod_congestion_state
Recommended Actions	The alert gets cleared when the system is back to normal state. For any additional guidance, contact My Oracle Support.

8.3.1.40 POD_DANGER_OF_CONGESTION

Table 8-41 POD_DANGER_OF_CONGESTION

Field	Details
Description	Pod Congestion status of {{$labels.service}} service is DoC
Summary	Pod Congestion status of {{$labels.service}} service is DoC
Severity	Major
Condition	The pod congestion status is set to Danger of Congestion.
OID	1.3.6.1.4.1.323.5.3.36.1.2.25
Metric Used	occnp_pod_congestion_state
Recommended Actions	The alert gets cleared when the system is back to normal state. For any additional guidance, contact My Oracle Support.

8.3.1.41 POD_PENDING_REQUEST_CONGESTED

Table 8-42 POD_PENDING_REQUEST_CONGESTED

Field	Details
Description	Pod Resource Congestion status of {{$labels.service}} service is congested for PendingRequest type.
Summary	Pod Resource Congestion status of {{$labels.service}} service is congested for PendingRequest type.
Severity	Critical
Condition	The pod congestion status is set to congested for PendingRequest.
OID	1.3.6.1.4.1.323.5.3.36.1.2.28
Metric Used	occnp_pod_resource_congestion_state{type="queue"}
Recommended Actions	The alert gets cleared when the pending requests in the queue comes below the configured threshold value. For any additional guidance, contact My Oracle Support.

8.3.1.42 POD_PENDING_REQUEST_DANGER_OF_CONGESTION

Table 8-43 POD_PENDING_REQUEST_DANGER_OF_CONGESTION

Field	Details
Description	Pod Resource Congestion status of {{$labels.service}} service is DoC for PendingRequest type.
Summary	Pod Resource Congestion status of {{$labels.service}} service is DoC for PendingRequest type.
Severity	Major
Condition	The pod congestion status is set to DoC for pending requests.
OID	1.3.6.1.4.1.323.5.3.36.1.2.27
Metric Used	occnp_pod_resource_congestion_state{type="queue"}
Recommended Actions	The alert gets cleared when the pending requests in the queue comes below the configured threshold value. For any additional guidance, contact My Oracle Support.

8.3.1.43 POD_CPU_CONGESTED

Table 8-44 POD_CPU_CONGESTED

Field	Details
Description	Pod Resource Congestion status of {{$labels.service}} service is congested for CPU type.
Summary	Pod Resource Congestion status of {{$labels.service}} service is congested for CPU type.
Severity	Critical
Condition	The pod congestion status is set to congested for CPU.
OID	1.3.6.1.4.1.323.5.3.36.1.2.30
Metric Used	occnp_pod_resource_congestion_state{type="cpu"}

Table 8-44 (Cont.) POD_CPU_CONGESTED

Field	Details
Recommended Actions	The alert gets cleared when the system CPU usage comes below the configured threshold value. For any additional guidance, contact My Oracle Support.

8.3.1.44 POD_CPU_DANGER_OF_CONGESTION

Table 8-45 POD_CPU_DANGER_OF_CONGESTION

Field	Details
Description	Pod Resource Congestion status of {{$labels.service}} service is DoC for CPU type.
Summary	Pod Resource Congestion status of {{$labels.service}} service is DoC for CPU type.
Severity	Major
Condition	The pod congestion status is set to DoC for CPU.
OID	1.3.6.1.4.1.323.5.3.36.1.2.29
Metric Used	occnp_pod_resource_congestion_state{type="cpu"}
Recommended Actions	The alert gets cleared when the system CPU usage comes below the configured threshold value. For any additional guidance, contact My Oracle Support.

8.3.1.45 SERVICE_OVERLOADED

Table 8-46 SERVICE_OVERLOADED

Field	Details
Description	Overload Level of {{$labels.service}} service is L1
Summary	Overload Level of {{$labels.service}} service is L1
Severity	Minor
Condition	The overload level of the service is L1.
OID	1.3.6.1.4.1.323.5.3.36.1.2.40
Metric Used	load_level
Recommended Actions	The alert gets cleared when the system is back to normal state. For any additional guidance, contact My Oracle Support.

Table 8-47 SERVICE_OVERLOADED

Field	Details
Description	Overload Level of {{$labels.service}} service is L2
Summary	Overload Level of {{$labels.service}} service is L2
Severity	Major
Condition	The overload level of the service is L2.
OID	1.3.6.1.4.1.323.5.3.36.1.2.40
Metric Used	load_level

Table 8-47 (Cont.) SERVICE_OVERLOADED

Field	Details
Recommended Actions	The alert gets cleared when the system is back to normal state. For any additional guidance, contact My Oracle Support.

Table 8-48 SERVICE_OVERLOADED

Field	Details
Description	Overload Level of {{$labels.service}} service is L3
Summary	Overload Level of {{$labels.service}} service is L3
Severity	Critical
Condition	The overload level of the service is L3.
OID	1.3.6.1.4.1.323.5.3.36.1.2.40
Metric Used	load_level
Recommended Actions	The alert gets cleared when the system is back to normal state. For any additional guidance, contact My Oracle Support.

8.3.1.46 SERVICE_RESOURCE_OVERLOADED

Alerts when service is in overload state due to memory usage

Table 8-49 SERVICE_RESOURCE_OVERLOADED

Field	Details
Description	{{$labels.service}} service is L1 for {{$labels.type}} type
Summary	{{$labels.service}} service is L1 for {{$labels.type}} type
Severity	Minor
Condition	The overload level of the service is L1 due to memory usage.
OID	1.3.6.1.4.1.323.5.3.36.1.2.41
Metric Used	service_resource_overload_level{type="memory"}
Recommended Actions	The alert gets cleared when the memory usage of the service is back to normal state. For any additional guidance, contact My Oracle Support.

Table 8-50 SERVICE_RESOURCE_OVERLOADED

Field	Details
Description	{{$labels.service}} service is L2 for {{$labels.type}} type
Summary	{{$labels.service}} service is L2 for {{$labels.type}} type
Severity	Major
Condition	The overload level of the service is L2 due to memory usage.
OID	1.3.6.1.4.1.323.5.3.36.1.2.41
Metric Used	service_resource_overload_level{type="memory"}
Recommended Actions	The alert gets cleared when the memory usage of the service is back to normal state. For any additional guidance, contact My Oracle Support.

Table 8-51 SERVICE_RESOURCE_OVERLOADED

Field	Details
Description	{{$labels.service}} service is L3 for {{$labels.type}} type.
Summary	{{$labels.service}} service is L3 for {{$labels.type}} type
Severity	Critical
Condition	The overload level of the service is L3 due to memory usage.
OID	1.3.6.1.4.1.323.5.3.36.1.2.41
Metric Used	service_resource_overload_level{type="memory"}
Recommended Actions	The alert gets cleared when the memory usage of the service is back to normal state. For any additional guidance, contact My Oracle Support.

Alerts when service is in overload state due to CPU usage

Table 8-52 SERVICE_RESOURCE_OVERLOADED

Field	Details
Description	{{$labels.service}} service is L1 for {{$labels.type}} type
Summary	{{$labels.service}} service is L1 for {{$labels.type}} type
Severity	Minor
Condition	The overload level of the service is L1 due to CPU usage.
OID	1.3.6.1.4.1.323.5.3.36.1.2.41
Metric Used	service_resource_overload_level{type="cpu"}
Recommended Actions	The alert gets cleared when the CPU usage of the service is back to normal state. For any additional guidance, contact My Oracle Support.

Table 8-53 SERVICE_RESOURCE_OVERLOADED

Field	Details
Description	{{$labels.service}} service is L2 for {{$labels.type}} type
Summary	{{$labels.service}} service is L2 for {{$labels.type}} type
Severity	Major
Condition	The overload level of the service is L2 due to CPU usage.
OID	1.3.6.1.4.1.323.5.3.36.1.2.41
Metric Used	service_resource_overload_level{type="cpu"}
Recommended Actions	The alert gets cleared when the CPU usage of the service is back to normal state. For any additional guidance, contact My Oracle Support.

Table 8-54 SERVICE_RESOURCE_OVERLOADED

Field	Details
Description	{{$labels.service}} service is L3 for {{$labels.type}} type
Summary	{{$labels.service}} service is L3 for {{$labels.type}} type
Severity	Major
Condition	The overload level of the service is L3 due to CPU usage.
OID	1.3.6.1.4.1.323.5.3.36.1.2.41

Table 8-54 (Cont.) SERVICE_RESOURCE_OVERLOADED

Field	Details
Metric Used	service_resource_overload_level{type="cpu"}
Recommended Actions	The alert gets cleared when the CPU usage of the service is back to normal state. For any additional guidance, contact My Oracle Support.

Alerts when service is in overload state due to number of pending messages

Table 8-55 SERVICE_RESOURCE_OVERLOADED

Field	Details
Description	{{$labels.service}} service is L1 for {{$labels.type}} type
Summary	{{$labels.service}} service is L1 for {{$labels.type}} type
Severity	Minor
Condition	The overload level of the service is L1 due to number of pending messages.
OID	1.3.6.1.4.1.323.5.3.36.1.2.41
Metric Used	service_resource_overload_level{type="svc_pending_count"}
Recommended Actions	The alert gets cleared when the number of pending messages of the service is back to normal state. For any additional guidance, contact My Oracle Support.

Table 8-56 SERVICE_RESOURCE_OVERLOADED

Field	Details
Description	{{$labels.service}} service is L2 for {{$labels.type}} type
Summary	{{$labels.service}} service is L2 for {{$labels.type}} type
Severity	Major
Condition	The overload level of the service is L2 due to number of pending messages.
OID	1.3.6.1.4.1.323.5.3.36.1.2.41
Metric Used	service_resource_overload_level{type="svc_pending_count"}
Recommended Actions	The alert gets cleared when the number of pending messages of the service is back to normal state. For any additional guidance, contact My Oracle Support.

Table 8-57 SERVICE_RESOURCE_OVERLOADED

Field	Details
Description	{{$labels.service}} service is L3 for {{$labels.type}} type
Summary	{{$labels.service}} service is L3 for {{$labels.type}} type
Severity	Critical
Condition	The overload level of the service is L3 due to number of pending messages.
OID	1.3.6.1.4.1.323.5.3.36.1.2.41
Metric Used	service_resource_overload_level{type="svc_pending_count"}
Recommended Actions	The alert gets cleared when the number of pending messages of the service is back to normal state. For any additional guidance, contact My Oracle Support.

Alerts when service is in overload state due to number of failed requests

Table 8-58 SERVICE_RESOURCE_OVERLOADED

Field	Details
Description	{{$labels.service}} service is L1 for {{$labels.type}} type.
Summary	{{$labels.service}} service is L1 for {{$labels.type}} type.
Severity	Minor
Condition	The overload level of the service is L1 due to number of failed requests.
OID	1.3.6.1.4.1.323.5.3.36.1.2.41
Metric Used	service_resource_overload_level{type="svc_failure_count"}
Recommended Actions	The alert gets cleared when the number of failed messages of the service is back to normal state. For any additional guidance, contact My Oracle Support.

Table 8-59 SERVICE_RESOURCE_OVERLOADED

Field	Details
Description	{{$labels.service}} service is L2 for {{$labels.type}} type.
Summary	{{$labels.service}} service is L2 for {{$labels.type}} type.
Severity	Major
Condition	The overload level of the service is L2 due to number of failed requests.
OID	1.3.6.1.4.1.323.5.3.36.1.2.41
Metric Used	service_resource_overload_level{type="svc_failure_count"}
Recommended Actions	The alert gets cleared when the number of failed messages of the service is back to normal state. For any additional guidance, contact My Oracle Support.

Table 8-60 SERVICE_RESOURCE_OVERLOADED

Field	Details
Description	{{$labels.service}} service is L3 for {{$labels.type}} type.
Summary	{{$labels.service}} service is L3 for {{$labels.type}} type.
Severity	Critical
Condition	The overload level of the service is L3 due to number of failed requests.
OID	1.3.6.1.4.1.323.5.3.36.1.2.41
Metric Used	service_resource_overload_level{type="svc_failure_count"}
Recommended Actions	The alert gets cleared when the number of failed messages of the service is back to normal state. For any additional guidance, contact My Oracle Support.

8.3.1.47 SUBSCRIBER_NOTIFICATION_ERROR_EXCEEDS_CRITICAL_THRESHOLD

Table 8-61 SUBSCRIBER_NOTIFICATION_ERROR_EXCEEDS_CRITICAL_THRESHOLD

Field	Details
Description	Notification Transaction Error exceeds the critical threshold limit for a given Subscriber Notification server
Summary	Transaction Error exceeds the critical threshold limit for a given Subscriber Notification server
Severity	Critical
Condition	The number of error responses for a given subscriber notification server exceeds the critical threshold of 1000.
OID	1.3.6.1.4.1.323.5.3.36.1.2.42
Metric Used	http_notification_response_total{responseCode!~"2.*"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

Table 8-62 SUBSCRIBER_NOTIFICATION_ERROR_EXCEEDS_MAJOR_THRESHOLD

Field	Details
Description	Notification Transaction Error exceeds the major threshold limit for a given Subscriber Notification server
Summary	Transaction Error exceeds the major threshold limit for a given Subscriber Notification server
Severity	Major
Condition	The number of error responses for a given subscriber notification server exceeds the major threshold value, that is, between 750 and 1000.
OID	1.3.6.1.4.1.323.5.3.36.1.2.42
Metric Used	http_notification_response_total{responseCode!~"2.*"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

Table 8-63 SUBSCRIBER_NOTIFICATION_ERROR_EXCEEDS_MINOR_THRESHOLD

Field	Details
Description	Notification Transaction Error exceeds the minor threshold limit for a given Subscriber Notification server
Summary	Transaction Error exceeds the minor threshold limit for a given Subscriber Notification server
Severity	Minor
Condition	The number of error responses for a given subscriber notification server exceeds the minor threshold value, that is, between 500 and 750.
OID	1.3.6.1.4.1.323.5.3.36.1.2.42
Metric Used	http_notification_response_total{responseCode!~"2.*"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.48 SYSTEM_IMPAIRMENT_MAJOR

Table 8-64 SYSTEM_IMPAIRMENT_MAJOR

Field	Details
Description	Major impairment alert raised for REPLICATION_FAILED or REPLICATION_CHANNEL_DOWN or BINLOG_STORAGE usage
Summary	Major impairment alert raised for REPLICATION_FAILED or REPLICATION_CHANNEL_DOWN or BINLOG_STORAGE usage
Severity	Major
Condition	Major Impairment alert
OID	1.3.6.1.4.1.323.5.3.36.1.2.43
Metric Used	db_tier_replication_status
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.49 SYSTEM_IMPAIRMENT_CRITICAL

Table 8-65 SYSTEM_IMPAIRMENT_CRITICAL

Field	Details
Description	Critical Impairment alert raised for REPLICATION_FAILED or REPLICATION_CHANNEL_DOWN or BINLOG_STORAGE usage
Summary	Critical Impairment alert raised for REPLICATION_FAILED or REPLICATION_CHANNEL_DOWN or BINLOG_STORAGE usage
Severity	Critical
Condition	Critical Impairment alert
OID	1.3.6.1.4.1.323.5.3.36.1.2.43
Metric Used	db_tier_replication_status
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.50 SYSTEM_OPERATIONAL_STATE_NORMAL

Table 8-66 SYSTEM_OPERATIONAL_STATE_NORMAL

Field	Details
Description	System Operational State is now in normal state
Summary	System Operational State is now in normal state
Severity	Info
Condition	System Operational State is now in normal state
OID	1.3.6.1.4.1.323.5.3.36.1.2.44
Metric Used	system_operational_state == 1
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.51 SYSTEM_OPERATIONAL_STATE_PARTIAL_SHUTDOWN

Table 8-67 SYSTEM_OPERATIONAL_STATE_PARTIAL_SHUTDOWN

Field	Details
Description	System Operational State is now in partial shutdown state.
Summary	System Operational State is now in partial shutdown state.
Severity	Info
Condition	System Operational State is now in partial shutdown state
OID	1.3.6.1.4.1.323.5.3.36.1.2.44
Metric Used	system_operational_state == 2
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.52 SYSTEM_OPERATIONAL_STATE_COMPLETE_SHUTDOWN

Table 8-68 SYSTEM_OPERATIONAL_COMPLETE_SHUTDOWN

Field	Details
Description	System Operational State is now in complete shutdown state
Summary	System Operational State is now in complete shutdown state
Severity	Info
Condition	System Operational State is now in complete shutdown state
OID	1.3.6.1.4.1.323.5.3.36.1.2.44
Metric Used	system_operational_state == 3
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.53 TDFConnectionDown

Table 8-69 TDFConnectionDown

Field	Details
Description	TDF connection is down.
Summary	TDF connection is down.
Severity	Critical
Condition	occnp_diam_conn_app_network{applicationName="Sd"} == 0
OID	1.3.6.1.4.1.323.5.3.52.1.2.48
Metric Used	occnp_diam_conn_app_network
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.54 DiamConnPeerDown

Table 8-70 DiamConnPeerDown

Field	Details
Description	Diameter connection to peer is down.
Summary	Diameter connection to peer is down.
Severity	Major
Condition	Diameter connection to peer is down.
OID	1.3.6.1.4.1.323.5.3.52.1.2.50
Metric Used	occnp_diam_conn_network
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.55 DiamConnNetworkDown

Table 8-71 DiamConnNetworkDown

Field	Details
Description	All the diameter network connections are down.
Summary	All the diameter network connections are down.
Severity	Critical
Condition	sum by (kubernetes_namespace)(occnp_diam_conn_network) == 0
OID	1.3.6.1.4.1.323.5.3.52.1.2.51
Metric Used	occnp_diam_conn_network
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.56 DiamConnBackendDown

Table 8-72 DiamConnBackendDown

Field	Details
Description	All the diameter backend connections are down.
Summary	All the diameter backend connections are down.
Severity	Critical
Condition	sum by (kubernetes_namespace)(occnp_diam_conn_backend) == 0
OID	1.3.6.1.4.1.323.5.3.52.1.2.52
Metric Used	occnp_diam_conn_network
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.57 PerfInfoActiveOverloadThresholdFetchFailed

Table 8-73 PerfInfoActiveOverloadThresholdFetchFailed

Field	Details
Description	The application fails to get the current active overload level threshold data.
Summary	The application fails to get the current active overload level threshold data.
Severity	Major
Condition	active_overload_threshold_fetch_failed == 1
OID	1.3.6.1.4.1.323.5.3.52.1.2.53
Metric Used	active_overload_threshold_fetch_failed
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.58 SLASYFailCountExceedsCritcalThreshold

Table 8-74 SLASYFailCountExceedsCritcalThreshold

Field	Details
Description	SLA Sy fail count exceeds the critical threshold limit
Summary	SLA Sy fail count exceeds the critical threshold limit
Severity	Critical
Condition	sum(rate(occnp_diam_response_local_total{msgType="SLA", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="SLA"}[5m])) * 100 > 90
OID	1.3.6.1.4.1.323.5.3.52.1.2.58
Metric Used	occnp_diam_response_local_total
Recommended Actions	Check the connectivity between diam-gw pod(s) and OCS server and ensure connectivity is present. If the user hasn't been added in the OCS configuration, then configure the user(s). For any additional guidance, contact My Oracle Support.

8.3.1.59 SLASYFailCountExceedsMajorThreshold

Table 8-75 SLASYFailCountExceedsMajorThreshold

Field	Details
Description	SLA Sy fail count exceeds the major threshold limit
Summary	SLA Sy fail count exceeds the major threshold limit
Severity	Major
Condition	sum(rate(occnp_diam_response_local_total{msgType="SLA", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="SLA"}[5m])) * 100 > 80 and sum(rate(occnp_diam_response_local_total{msgType="SLA", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="SLA"}[5m])) * 100 <= 90
OID	1.3.6.1.4.1.323.5.3.52.1.2.58
Metric Used	occnp_diam_response_local_total

Table 8-75 (Cont.) SLASYFailCountExceedsMajorThreshold

Field	Details
Recommended Actions	Check the connectivity between diam-gw pod(s) and OCS server and ensure connectivity is present. If the user hasn't been added in the OCS configuration, then configure the user(s). For any additional guidance, contact My Oracle Support.

8.3.1.60 SLASYFailCountExceedsMinorThreshold

Table 8-76 SLASYFailCountExceedsMinorThreshold

Field	Details
Description	SLA Sy fail count exceeds the minor threshold limit
Summary	SLA Sy fail count exceeds the minor threshold limit
Severity	Minor
Condition	sum(rate(occnp_diam_response_local_total{msgType="SLA", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="SLA"}[5m])) * 100 > 60 and sum(rate(occnp_diam_response_local_total{msgType="SLA", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="SLA"}[5m])) * 100 <= 80
OID	1.3.6.1.4.1.323.5.3.52.1.2.58
Metric Used	occnp_diam_response_local_total
Recommended Actions	Check the connectivity between diam-gw pod(s) and OCS server and ensure connectivity is present. If the user hasn't been added in the OCS configuration, then configure the user(s). For any additional guidance, contact My Oracle Support.

8.3.1.61 STASYFailCountExceedsCritcalThreshold

Table 8-77 STASYFailCountExceedsCritcalThreshold

Field	Details
Description	STA Sy fail count exceeds the critical threshold limit.
Summary	STA Sy fail count exceeds the critical threshold limit.
Severity	Critical
Condition	The failure rate of Sy STA responses is more than 90% of the total responses.
Expression	sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777302", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777302"}[5m])) * 100 > 90
OID	1.3.6.1.4.1.323.5.3.52.1.2.59
Metric Used	occnp_diam_response_local_total
Recommended Actions	Check the connectivity between diam-gw pod(s) and OCS server and ensure connectivity is present. If the user hasn't been added in the OCS configuration, then configure the user(s). For any additional guidance, contact My Oracle Support.

8.3.1.62 STA_SY_FAIL_COUNT_EXCEEDS_MAJOR_THRESHOLD

Table 8-78 STA_SY_FAIL_COUNT_EXCEEDS_MAJOR_THRESHOLD

Field	Details
Description	STA Sy fail count exceeds the major threshold limit.
Summary	STA Sy fail count exceeds the major threshold limit.
Severity	Major
Condition	The failure rate of Sy STA responses is more than 80% and less and or equal to 90% of the total responses.
Expression	sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777302", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777302"}[5m])) * 100 > 80 and sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777302", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777302"}[5m])) * 100 <= 90
OID	1.3.6.1.4.1.323.5.3.52.1.2.59
Metric Used	occnp_diam_response_local_total
Recommended Actions	Check the connectivity between diam-gw pod(s) and OCS server and ensure connectivity is present. If the user hasn't been added in the OCS configuration, then configure the user(s). For any additional guidance, contact My Oracle Support.

8.3.1.63 STASYFailCountExceedsMinorThreshold

Table 8-79 STASYFailCountExceedsMinorThreshold

Field	Details
Description	STA Sy fail count exceeds the minor threshold limit.
Summary	STA Sy fail count exceeds the minor threshold limit.
Severity	Minor
Condition	The failure rate of Sy STA responses is more than 60% and less and or equal to 80% of the total responses.
Expression	sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777302", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777302"}[5m])) * 100 > 60 and sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777302", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777302"}[5m])) * 100 <= 80
OID	1.3.6.1.4.1.323.5.3.52.1.2.59
Metric Used	occnp_diam_response_local_total
Recommended Actions	Check the connectivity between diam-gw pod(s) and OCS server and ensure connectivity is present. If the user hasn't been added in the OCS configuration, then configure the user(s). For any additional guidance, contact My Oracle Support.

8.3.1.64 SMSC_CONNECTION_DOWN

Table 8-80 STASYFailCountExceedsCritcalThreshold

Field	Details
Description	This alert is triggered when connection to SMSC host is down.
Summary	Connection to SMSC peer {{$labels.smscName}} is down in notifier service pod {{$labels.pod}}
Severity	Major
Condition	sum by(namespace, pod, smscName)(occnp_active_smsc_conn_count) == 0
OID	1.3.6.1.4.1.323.5.3.52.1.2.63
Metric Used	occnp_active_smsc_conn_count
Recommended Actions	Check the connectivity between diam-gw pod(s) and OCS server and ensure connectivity is present. If the user hasn't been added in the OCS configuration, then configure the user(s). For any additional guidance, contact My Oracle Support.

8.3.1.65 STA_RX_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Table 8-81 STASYFailCountExceedsCritcalThreshold

Field	Details
Description	STA Rx fail count exceeds the critical threshold limit.
Summary	STA Rx fail count exceeds the critical threshold limit.
Severity	Critical
Condition	The failure rate of Rx STA responses is more than 90% of the total responses.
Expression	sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777236", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777236"}[5m])) * 100 > 90
OID	1.3.6.1.4.1.323.5.3.52.1.2.64
Metric Used	occnp_diam_response_local_total{msgType="STA", appId="16777236", responseCode!~"2.*"}
Recommended Actions	Check the connectivity between diam-gw pod(s) and OCS server and ensure connectivity is present. Check that the session and user hasn't been removed in the OCS configuration, then configure the user(s). For any additional guidance, contact My Oracle Support.

8.3.1.66 STA_RX_FAIL_COUNT_EXCEEDS_MAJOR_THRESHOLD

Table 8-82 STA_RX_FAIL_COUNT_EXCEEDS_MAJOR_THRESHOLD

Field	Details
Description	STA Rx fail count exceeds the major threshold limit.
Summary	STA Rx fail count exceeds the major threshold limit.

Table 8-82 (Cont.) STA_RX_FAIL_COUNT_EXCEEDS_MAJOR_THRESHOLD

Field	Details
Severity	Major
Condition	The failure rate of Rx STA responses is more than 80% and less and or equal to 90% of the total responses.
Expression	sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777236", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777236"}[5m])) * 100 > 80 and sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777236", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777236"}[5m])) * 100 <= 90
OID	1.3.6.1.4.1.323.5.3.52.1.2.64
Metric Used	occnp_diam_response_local_total{msgType="STA", appId="16777236", responseCode!~"2.*"}
Recommended Actions	Check the connectivity between diam-gw pod(s) & AF and ensure connectivity is present. Check that the session and user is valid and hasn't been removed in the Policy database, then configure the user(s). For any additional guidance, contact My Oracle Support.

8.3.1.67 STA_RX_FAIL_COUNT_EXCEEDS_MINOR_THRESHOLD

Table 8-83 STA_RX_FAIL_COUNT_EXCEEDS_MINOR_THRESHOLD

Field	Details
Description	STA Rx fail count exceeds the minor threshold limit.
Summary	STA Rx fail count exceeds the minor threshold limit.
Severity	Minor
Condition	The failure rate of Rx STA responses is more than 60% and less and or equal to 80% of the total responses.
Expression	sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777236", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777236"}[5m])) * 100 > 60 and sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777236", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="STA", appId="16777236"}[5m])) * 100 <= 80
OID	1.3.6.1.4.1.323.5.3.52.1.2.64
Metric Used	occnp_diam_response_local_total{msgType="STA", appId="16777236", responseCode!~"2.*"}
Recommended Actions	Check the connectivity between diam-gw pod(s) & AF and ensure connectivity is present. Check that the session and user is valid and hasn't been removed in the Policy database, then configure the user(s). For any additional guidance, contact My Oracle Support.

8.3.1.68 SNA_SY_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Table 8-84 SNA_SY_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Field	Details
Description	SNA Sy fail count exceeds the critical threshold limit
Summary	SNA Sy fail count exceeds the critical threshold limit
Severity	Critical
Condition	The failure rate of Sy SNA responses is more than 90% of the total responses.
Expression	sum(rate(occnp_diam_response_local_total{msgType="SNA", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="SNA"}[5m])) * 100 > 90
OID	1.3.6.1.4.1.323.5.3.52.1.2.65
Metric Used	occnp_diam_response_local_total{msgType="SNA", responseCode!~"2.*"}
Recommended Actions	Check the connectivity between diam-gw pod(s) and OCS server and ensure connectivity is present. Check that the session and user hasn't been removed in the OCS configuration, then configure the user(s). For any additional guidance, contact My Oracle Support.

8.3.1.69 SNA_SY_FAIL_COUNT_EXCEEDS_MAJOR_THRESHOLD

Table 8-85 SNA_SY_FAIL_COUNT_EXCEEDS_MAJOR_THRESHOLD

Field	Details
Description	SNA Sy fail count exceeds the major threshold limit
Summary	SNA Sy fail count exceeds the major threshold limit
Severity	Major
Condition	The failure rate of Sy SNA responses is more than 80% and less and or equal to 90% of the total responses.
Expression	sum(rate(occnp_diam_response_local_total{msgType="SNA", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="SNA"}[5m])) * 100 > 80 and sum(rate(occnp_diam_response_local_total{msgType="SNA", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="SNA"}[5m])) * 100 <= 90
OID	1.3.6.1.4.1.323.5.3.52.1.2.65
Metric Used	occnp_diam_response_local_total{msgType="SNA", responseCode!~"2.*"}
Recommended Actions	Check the connectivity between diam-gw pod(s) and OCS server and ensure connectivity is present. Check that the session and user hasn't been removed in the OCS configuration, then configure the user(s). For any additional guidance, contact My Oracle Support.

8.3.1.70 SNA_SY_FAIL_COUNT_EXCEEDS_MINOR_THRESHOLD

Table 8-86 SNA_SY_FAIL_COUNT_EXCEEDS_MINOR_THRESHOLD

Field	Details
Description	SNA Sy fail count exceeds the minor threshold limit
Summary	SNA Sy fail count exceeds the minor threshold limit
Severity	Minor
Condition	The failure rate of Sy STA responses is more than 60% and less and or equal to 80% of the total responses.
Expression	sum(rate(occnp_diam_response_local_total{msgType="SNA", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="SNA"}[5m])) * 100 > 60 and sum(rate(occnp_diam_response_local_total{msgType="SNA", responseCode!~"2.*"}[5m])) / sum(rate(occnp_diam_response_local_total{msgType="SNA"}[5m])) * 100 <= 80
OID	1.3.6.1.4.1.323.5.3.52.1.2.65
Metric Used	occnp_diam_response_local_total{msgType="SNA", responseCode!~"2.*"}
Recommended Actions	Check the connectivity between diam-gw pod(s) and OCS server and ensure connectivity is present. Check that the session and user hasn't been removed in the OCS configuration, then configure the user(s). For any additional guidance, contact My Oracle Support.

8.3.1.71 STALE_DIAMETER_REQUEST_CLEANUP_MINOR

Table 8-87 STALE_DIAMETER_REQUEST_CLEANUP_MINOR

Field	Details
Description	This alerts is triggered when more than 10 % of the received Diameter requests are cancelled due to them being stale (received too late, or took too much time to process them).
Summary	
Severity	Minor
Expression	
OID	
Metric Used	• ocpm_stale_diam_request_cleanup_total • occnp_diam_request_local_total
Recommended Actions	

8.3.1.72 STALE_DIAMETER_REQUEST_CLEANUP_MAJOR

Table 8-88 STALE_DIAMETER_REQUEST_CLEANUP_MAJOR

Field	Details
Description	This alert is triggered when more than 20 % of the received Diameter requests are cancelled due to them being stale (received too late, or took too much time to process them).

Table 8-88 (Cont.) STALE_DIAMETER_REQUEST_CLEANUP_MAJOR

Field	Details
Summary	
Severity	Major
Expression	
OID	
Metric Used	• ocpm_late_arrival_rejection_total • occnp_diam_request_local_total
Recommended Actions	

8.3.1.73 STALE_DIAMETER_REQUEST_CLEANUP_CRITICAL

Table 8-89 STALE_DIAMETER_REQUEST_CLEANUP_CRITICAL

Field	Details
Description	This alert is triggered when more than 30 % of the received Diameter requests are cancelled due to them being stale (received too late, or took too much time to process them).
Summary	
Severity	Critical
Expression	
OID	
Metric Used	• ocpm_late_arrival_rejection_total • occnp_diam_request_local_total
Recommended Actions	

8.3.1.74 DIAM_GATEWAY_CERTIFICATE_EXPIRY_MINOR

Table 8-90 DIAM_GATEWAY_CERTIFICATE_EXPIRY_MINOR

Field	Details
Description	Certificate expiry in less than 6 months.
Summary	Certificate expiry in less than 6 months.
Severity	Minor
Condition	dgw_tls_cert_expiration_seconds - time() <= 15724800
OID	1.3.6.1.4.1.323.5.3.52.1.2.75
Metric Used	dgw_tls_cert_expiration_seconds
Recommended Actions	For any additional guidance, contact My Oracle Support (https://support.oracle.com).

8.3.1.75 DIAM_GATEWAY_CERTIFICATE_EXPIRY_MAJOR

Table 8-91 DIAM_GATEWAY_CERTIFICATE_EXPIRY_MAJOR

Field	Details
Description	Certificate expiry in less than 3 months.
Summary	Certificate expiry in less than 3 months.
Severity	Major
Condition	dgw_tls_cert_expiration_seconds - time() <= 7862400
OID	1.3.6.1.4.1.323.5.3.52.1.2.75
Metric Used	dgw_tls_cert_expiration_seconds
Recommended Actions	For any additional guidance, contact My Oracle Support (https://support.oracle.com).

8.3.1.76 DIAM_GATEWAY_CERTIFICATE_EXPIRY_CRITICAL

Table 8-92 DIAM_GATEWAY_CERTIFICATE_EXPIRY_CRITICAL

Field	Details
Description	Certificate expiry in less than 1 month.
Summary	Certificate expiry in less than 1 month.
Severity	Critical
Condition	dgw_tls_cert_expiration_seconds - time() <= 2592000
OID	1.3.6.1.4.1.323.5.3.52.1.2.75
Metric Used	dgw_tls_cert_expiration_seconds
Recommended Actions	For any additional guidance, contact My Oracle Support (https://support.oracle.com).

8.3.1.77 DGW_TLS_CONNECTION_FAILURE

Table 8-93 DGW_TLS_CONNECTION_FAILURE

Field	Details
Description	Alert for TLS connection establishment.
Summary	TLS Connection failure when Diam gateway is an initiator.
Severity	Major
Condition	sum by (namespace,reason) (occnp_diam_failed_conn_network) > 0
OID	1.3.6.1.4.1.323.5.3.52.1.2.81
Metric Used	occnp_diam_failed_conn_network
Recommended Actions	For any additional guidance, contact My Oracle Support (https://support.oracle.com).

8.3.1.78 POLICY_CONNECTION_FAILURE

Table 8-94 BSF_CONNECTION_FAILURE

Field	Details
Description	Connection failure on Egress and Ingress Gateways for incoming and outgoing connections.
Summary	
Severity	Major
Condition	This alert is raised when the TLS certificate is about to expire in three months.
OID	1.3.6.1.4.1.323.5.3.52.1.2.43
Metric Used	occnp_oc_ingressgateway_connection_failure_total
Recommended Actions	For any additional guidance, contact My Oracle Support (https://support.oracle.com).

8.3.1.79 DIAM_GATEWAY_CERTIFICATE_EXPIRY_CRITICAL

Table 8-95 DIAM_GATEWAY_CERTIFICATE_EXPIRY_CRITICAL

Field	Details
Description	TLS certificate to expire in 1 month.
Summary	security_cert_x509_expiration_seconds - time() <= 2592000
Severity	Critical
Condition	This alert is raised when the TLS certificate is about to expire in one month.
OID	1.3.6.1.4.1.323.5.3.52.1.2.44
Metric Used	security_cert_x509_expiration_seconds
Recommended Actions	For any additional guidance, contact My Oracle Support (https://support.oracle.com).

8.3.1.80 DIAM_GATEWAY_CERTIFICATE_EXPIRY_MAJOR

Table 8-96 DIAM_GATEWAY_CERTIFICATE_EXPIRY_MAJOR

Field	Details
Description	TLS certificate to expire in 3 months.
Summary	security_cert_x509_expiration_seconds - time() <= 7862400
Severity	Major
Condition	This alert is raised when the TLS certificate is about to expire in three months.
OID	1.3.6.1.4.1.323.5.3.52.1.2.44
Metric Used	security_cert_x509_expiration_seconds

Table 8-96 (Cont.) DIAM_GATEWAY_CERTIFICATE_EXPIRY_MAJOR

Field	Details
Recommended Actions	For any additional guidance, contact My Oracle Support (https://support.oracle.com).

8.3.1.81 DIAM_GATEWAY_CERTIFICATE_EXPIRY_MINOR

Table 8-97 DIAM_GATEWAY_CERTIFICATE_EXPIRY_MINOR

Field	Details
Description	TLS certificate to expire in 6 months.
Summary	security_cert_x509_expiration_seconds - time() <= 15724800
Severity	Minor
Condition	This alert is raised when the TLS certificate is about to expire in six months.
OID	1.3.6.1.4.1.323.5.3.52.1.2.44
Metric Used	security_cert_x509_expiration_seconds
Recommended Actions	For any additional guidance, contact My Oracle Support (https://support.oracle.com).

8.3.1.82 AUDIT_NOT_RUNNING

Table 8-98 AUDIT_NOT_RUNNING

Field	Details
Description	Audit has not been running for at least 1 hour.
Summary	Audit has not been running for at least 1 hour.
Severity	CRITICAL
Condition	(absent_over_time(spring_data_repository_invocations_seconds_count{method="getQueuedTablesToAudit"}[1h]) == 1) OR (sum(increase(spring_data_repository_invocations_seconds_count{method="getQueuedTablesToAudit"}[1h])) == 0)
OID	1.3.6.1.4.1.323.5.3.52.1.2.78
Metric Used	spring_data_repository_invocations_seconds_count
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.83 DIAMETER_POD_ERROR_RESPONSE_MINOR

Table 8-99 DIAMETER_POD_ERROR_RESPONSE_MINOR

Field	Details
Description	At least 1% of the Diam Response connection requests failed with error DIAMETER_UNABLE_TO_DELIVER.
Summary	At least 1% of the Diam Response connection requests failed with error DIAMETER_UNABLE_TO_DELIVER.

Table 8-99 (Cont.) DIAMETER_POD_ERROR_RESPONSE_MINOR

Field	Details
Severity	MINOR
Condition	(topk(1,((sort_desc(sum by (pod) (rate(ocbsf_diam_response_network_total{responseCode="3002"}[2m])))/ (sum by (pod) (rate(ocbsf_diam_response_network_total[2m]))) * 100))) >=1
OID	1.3.6.1.4.1.323.5.3.52.1.2.79
Metric Used	ocbsf_diam_response_network_total
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.84 LOCK_ACQUISITION_EXCEEDS_MAJOR_THRESHOLD

Table 8-100 DIAMETER_POD_ERROR_RESPONSE_MAJOR

Field	Details
Description	At least 5% of the Diam Response connection requests failed with error DIAMETER_UNABLE_TO_DELIVER.
Summary	At least 5% of the Diam Response connection requests failed with error DIAMETER_UNABLE_TO_DELIVER.
Severity	MAJOR
Condition	(topk(1,((sort_desc(sum by (pod) (rate(ocbsf_diam_response_network_total{responseCode="3002"}[2m])))/ (sum by (pod) (rate(ocbsf_diam_response_network_total[2m]))) * 100))) >=5
OID	1.3.6.1.4.1.323.5.3.52.1.2.79
Metric Used	ocbsf_diam_response_network_total
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.85 DIAMETER_POD_ERROR_RESPONSE_CRITICAL

Table 8-101 DIAMETER_POD_ERROR_RESPONSE_CRITICAL

Field	Details
Description	At least 10% of the Diam Response connection requests failed with error DIAMETER_UNABLE_TO_DELIVER
Summary	At least 10% of the Diam Response connection requests failed with error DIAMETER_UNABLE_TO_DELIVER
Severity	CRITICAL
Condition	(topk(1,((sort_desc(sum by (pod) (rate(ocbsf_diam_response_network_total{responseCode="3002"}[2m])))/ (sum by (pod) (rate(ocbsf_diam_response_network_total[2m]))) * 100))) >=10
OID	1.3.6.1.4.1.323.5.3.52.1.2.79
Metric Used	ocbsf_diam_response_network_total
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.1.86 LOCK_ACQUISITION_EXCEEDS_CRITICAL_THRESHOLD

Table 8-102 LOCK_ACQUISITION_EXCEEDS_CRITICAL_THRESHOLD

Field	Details
Description	The lock requests fails to acquire the lock count exceeds the critical threshold limit. The (current value is: {{ $value }})
Summary	Keys used in Bulwark lock request which are already in locked state detected above 75 Percent of Total Transactions.
Severity	Critical
Expression	(sum by (namespace) (increase(lock_response_total{requestType="acquireLock",responseType="failure"} [5m])) /sum by (namespace) (increase(lock_request_total{requestType="acquireLock"} [5m]))) * 100 >=75
OID	1.3.6.1.4.1.323.5.3.52.1.2.69
Metric Used	
Recommended Actions	

8.3.1.87 LOCK_ACQUISITION_EXCEEDS_MAJOR_THRESHOLD

Table 8-103 LOCK_ACQUISITION_EXCEEDS_MAJOR_THRESHOLD

Field	Details
Description	The lock requests fails to acquire the lock count exceeds the major threshold limit. The (current value is: {{ $value }})
Summary	Keys used in Bulwark lock request which are already in locked state detected above 50 Percent of Total Transactions.
Severity	Major
Expression	(sum by (namespace) (increase(lock_response_total{requestType="acquireLock",responseType="failure"} [5m])) /sum by (namespace) (increase(lock_request_total{requestType="acquireLock"} [5m]))) * 100 >= 50 < 75
OID	1.3.6.1.4.1.323.5.3.52.1.2.69
Metric Used	
Recommended Actions	

8.3.1.88 LOCK_ACQUISITION_EXCEEDS_MINOR_THRESHOLD

Table 8-104 LOCK_ACQUISITION_EXCEEDS_MINOR_THRESHOLD

Field	Details
Description	The lock requests fails to acquire the lock count exceeds the minor threshold limit. The (current value is: {{ $value }})
Summary	Keys used in Bulwark lock request which are already in locked state detected above 20 Percent of Total Transactions.
Severity	Minor

Table 8-104 (Cont.) LOCK_ACQUISITION_EXCEEDS_MINOR_THRESHOLD

Field	Details
Expression	(sum by (namespace) (increase(lock_response_total{requestType="acquireLock",responseType="failure"} [5m])) /sum by (namespace) (increase(lock_request_total{requestType="acquireLock"} [5m]))) * 100 >=20 < 50
OID	1.3.6.1.4.1.323.5.3.52.1.2.69
Metric Used	
Recommended Actions	

8.3.1.89 CERTIFICATE_EXPIRY_MINOR

Table 8-105 CERTIFICATE_EXPIRY_MINOR

Field	Details
Description	Certificate expiry in less than 6 months
Summary	Certificate expiry in less than 6 months
Severity	MINOR
Condition	security_cert_x509_expiration_seconds - time() <= 15724800
OID	1.3.6.1.4.1.323.5.3.52.1.2.77
Metric Used	-
Recommended Actions	-

8.3.1.90 CERTIFICATE_EXPIRY_MAJOR

Table 8-106 CERTIFICATE_EXPIRY_MAJOR

Field	Details
Description	Certificate expiry in less than 3 months
Summary	Certificate expiry in less than 3 months
Severity	MAJOR
Condition	security_cert_x509_expiration_seconds - time() <= 7862400
OID	1.3.6.1.4.1.323.5.3.52.1.2.77
Metric Used	-
Recommended Actions	-

8.3.1.91 CERTIFICATE_EXPIRY_CRITICAL

Table 8-107 CERTIFICATE_EXPIRY_CRITICAL

Field	Details
Description	Certificate expiry in less than 1 months
Summary	Certificate expiry in less than 1 months
Severity	CRITICAL
Condition	security_cert_x509_expiration_seconds - time() <= 2592000

Table 8-107 (Cont.) CERTIFICATE_EXPIRY_CRITICAL

Field	Details
OID	1.3.6.1.4.1.323.5.3.52.1.2.77
Metric Used	-
Recommended Actions	-

8.3.1.92 PERF_INFO_ACTIVE_OVERLOADTHRESHOLD_DATA_PRESENT

Table 8-108 PERF_INFO_ACTIVE_OVERLOADTHRESHOLD_DATA_PRESENT

Field	Details
Description	
Summary	
Severity	MINOR
Condition	active_overload_threshold_fetch_failed == 1
OID	1.3.6.1.4.1.323.5.3.52.1.2.53
Metric Used	
Recommended Actions	

8.3.1.93 UDR_C_STALE_HTTP_REQUEST_CLEANUP_MINOR

Table 8-109 UDR_C_STALE_HTTP_REQUEST_CLEANUP_MINOR

Field	Details
Description	More than 10% of incoming requests towards UDR-connector is rejected due to request being stale on arrival or during processing by the connector
Summary	More than 10% of incoming requests towards UDR-connector is rejected due to request being stale on arrival or during processing by the connector
Severity	MINOR
Condition	(sum by (namespace) (rate(occnp_late_processing_rejection_total{mode="UDR-C"}[5m])) + sum by (namespace) (rate(occnp_late_arrival_rejection_total{mode="UDR-C"}[5m])))/(sum by (namespace) (rate(ocpm_userservice_inbound_count_total{service_resource="udr-service"}[5m])) + sum by (namespace) (rate(occnp_late_arrival_rejection_total{mode="UDR-C"}[5m]))) * 100 > 10
OID	1.3.6.1.4.1.323.5.3.52.1.2.85
Metric Used	-
Recommended Actions	-

8.3.1.94 UDR_C_STALE_HTTP_REQUEST_CLEANUP_MAJOR

Table 8-110 UDR_C_STALE_HTTP_REQUEST_CLEANUP_MAJOR

Field	Details
Description	More than 20% of incoming requests towards UDR-connector is rejected due to request being stale on arrival or during processing by the connector

Table 8-110 (Cont.) UDR_C_STALE_HTTP_REQUEST_CLEANUP_MAJOR

Field	Details
Summary	More than 20% of incoming requests towards UDR-connector is rejected due to request being stale on arrival or during processing by the connector
Severity	MAJOR
Condition	(sum by (namespace) (rate(occnp_late_processing_rejection_total{mode="UDR-C"}[5m])) + sum by (namespace) (rate(occnp_late_arrival_rejection_total{mode="UDR-C"}[5m])))/(sum by (namespace) (rate(ocpm_userservice_inbound_count_total{service_resource="udr-service"}[5m])) + sum by (namespace) (rate(occnp_late_arrival_rejection_total{mode="UDR-C"}[5m]))) * 100 > 20
OID	1.3.6.1.4.1.323.5.3.52.1.2.85
Metric Used	-
Recommended Actions	-

8.3.1.95 UDR_C_STALE_HTTP_REQUEST_CLEANUP_CRITICAL

Table 8-111 UDR_C_STALE_HTTP_REQUEST_CLEANUP_CRITICAL

Field	Details
Description	More than 30% of incoming requests towards UDR-connector is rejected due to request being stale on arrival or during processing by the connector
Summary	More than 30% of incoming requests towards UDR-connector is rejected due to request being stale on arrival or during processing by the connector
Severity	CRITICAL
Condition	(sum by (namespace) (rate(occnp_late_processing_rejection_total{mode="UDR-C"}[5m])) + sum by (namespace) (rate(occnp_late_arrival_rejection_total{mode="UDR-C"}[5m])))/(sum by (namespace) (rate(ocpm_userservice_inbound_count_total{service_resource="udr-service"}[5m])) + sum by (namespace) (rate(occnp_late_arrival_rejection_total{mode="UDR-C"}[5m]))) * 100 > 30
OID	1.3.6.1.4.1.323.5.3.52.1.2.85
Metric Used	-
Recommended Actions	-

8.3.1.96 CHF_C_STALE_HTTP_REQUEST_CLEANUP_MINOR

Table 8-112 CHF_C_STALE_HTTP_REQUEST_CLEANUP_MINOR

Field	Details
Description	More than 10% of incoming requests towards CHF-connector is rejected due to request being stale on arrival or during processing by the connector
Summary	More than 10% of incoming requests towards CHF-connector is rejected due to request being stale on arrival or during processing by the connector
Severity	MINOR

Table 8-112 (Cont.) CHF_C_STALE_HTTP_REQUEST_CLEANUP_MINOR

Field	Details
Condition	(sum by (namespace) (rate(occnp_late_processing_rejection_total{mode="CHF-C"}[5m])) + sum by (namespace) (rate(occnp_late_arrival_rejection_total{mode="CHF-C"}[5m])))/(sum by (namespace) (rate(ocpm_userservice_inbound_count_total{service_resource="chf-service"}[5m])) + sum by (namespace) (rate(occnp_late_arrival_rejection_total{mode="CHF-C"}[5m]))) * 100 > 10
OID	1.3.6.1.4.1.323.5.3.52.1.2.86
Metric Used	-
Recommended Actions	-

8.3.1.97 CHF_C_STALE_HTTP_REQUEST_CLEANUP_MAJOR

Table 8-113 CHF_C_STALE_HTTP_REQUEST_CLEANUP_MAJOR

Field	Details
Description	More than 20% of incoming requests towards CHF-connector is rejected due to request being stale on arrival or during processing by the connector
Summary	More than 20% of incoming requests towards CHF-connector is rejected due to request being stale on arrival or during processing by the connector
Severity	MAJOR
Condition	(sum by (namespace) (rate(occnp_late_processing_rejection_total{mode="CHF-C"}[5m])) + sum by (namespace) (rate(occnp_late_arrival_rejection_total{mode="CHF-C"}[5m])))/(sum by (namespace) (rate(ocpm_userservice_inbound_count_total{service_resource="chf-service"}[5m])) + sum by (namespace) (rate(occnp_late_arrival_rejection_total{mode="CHF-C"}[5m]))) * 100 > 20
OID	1.3.6.1.4.1.323.5.3.52.1.2.86
Metric Used	-
Recommended Actions	-

8.3.1.98 CHF_C_STALE_HTTP_REQUEST_CLEANUP_CRITICAL

Table 8-114 CHF_C_STALE_HTTP_REQUEST_CLEANUP_CRITICAL

Field	Details
Description	More than 30% of incoming requests towards CHF-connector is rejected due to request being stale on arrival or during processing by the connector
Summary	More than 30% of incoming requests towards CHF-connector is rejected due to request being stale on arrival or during processing by the connector
Severity	CRITICAL
Condition	(sum by (namespace) (rate(occnp_late_processing_rejection_total{mode="CHF-C"}[5m])) + sum by (namespace) (rate(occnp_late_arrival_rejection_total{mode="CHF-C"}[5m])))/(sum by (namespace) (rate(ocpm_userservice_inbound_count_total{service_resource="chf-service"}[5m])) + sum by (namespace) (rate(occnp_late_arrival_rejection_total{mode="CHF-C"}[5m]))) * 100 > 30
OID	1.3.6.1.4.1.323.5.3.52.1.2.86

Table 8-114 (Cont.) CHF_C_STALE_HTTP_REQUEST_CLEANUP_CRITICAL

Field	Details
Metric Used	-
Recommended Actions	-

8.3.1.99 EGRESS_GATEWAY_DD_UNREACHABLE_MAJOR

Table 8-115 EGRESS_GATEWAY_DD_UNREACHABLE_MAJOR

Field	Details
Description	This alarm is raised when OCNADD is not reachable.
Summary	'kubernetes_namespace: {{$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }} BSF Egress Gateway Data Director unreachable'
Severity	Major
Condition	This alarm is raised when data director is not reachable from Egress Gateway.
OID	1.3.6.1.4.1.323.5.3.37.1.2.48
Metric Used	oc_egressgateway_dd_unreachable
Recommended Actions	Alert gets cleared automatically when the connection with data director is established.

8.3.1.100 INGRESS_GATEWAY_DD_UNREACHABLE_MAJOR

Table 8-116 INGRESS_GATEWAY_DD_UNREACHABLE_MAJOR

Field	Details
Description	This alarm is raised when OCNADD is not reachable.
Summary	'kubernetes_namespace: {{$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }} BSF Ingress Gateway Data Director unreachable'
Severity	Major
Condition	This alarm is raised when data director is not reachable from Ingress Gateway.
OID	1.3.6.1.4.1.323.5.3.37.1.2.47
Metric Used	oc_ingressgateway_dd_unreachable
Recommended Actions	Alert gets cleared automatically when the connection with data director is established.

8.3.2 PCF Alerts

This section provides information on PCF alerts.

8.3.2.1 INGRESS_ERROR_RATE_ABOVE_10_PERCENT_PER_POD

Table 8-117 INGRESS_ERROR_RATE_ABOVE_10_PERCENT_PER_POD

Field	Details
Description	Ingress Error Rate above 10 Percent in {{$labels.kubernetes_name}} in {{$labels.kubernetes_namespace}}
Summary	Transaction Error Rate in {{$labels.kubernetes_node}} (current value is: {{ $value }})
Severity	Critical
Condition	The total number of failed transactions per pod is above 10 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.36.1.2.2
Metric Used	ocpm_ingress_response_total
Recommended Actions	The alert gets cleared when the number of failed transactions are below 10% of the total transactions. To assess the reason for failed transactions, perform the following steps: 1. Check the service specific metrics to understand the service specific errors. 2. The service specific errors can be further filtered for errors specific to a method such as GET, PUT, POST, DELETE, and PATCH. For any additional guidance, contact My Oracle Support.

8.3.2.2 SM_TRAFFIC_RATE_ABOVE_THRESHOLD

Table 8-118 SM_TRAFFIC_RATE_ABOVE_THRESHOLD

Field	Details
Description	SM service Ingress traffic Rate is above threshold of Max MPS (current value is: {{ $value }})
Summary	Traffic Rate is above 90 Percent of Max requests per second
Severity	Major
Condition	The total SM service Ingress traffic rate has crossed the configured threshold of 900 TPS. Default value of this alert trigger point in PCF_Alertrules.yaml file is when SM service Ingress Rate crosses 90% of maximum ingress requests per second.
OID	1.3.6.1.4.1.323.5.3.36.1.2.3
Metric Used	ocpm_ingress_request_total{servicename_3gpp="npcf-smpolicycontrol"}
Recommended Actions	The alert gets cleared when the Ingress traffic rate falls below the threshold. **Note**: Threshold levels can be configured using the `PCF_Alertrules.yaml` file. It is recommended to assess the reason for additional traffic. Perform the following steps to analyze the cause of increased traffic: 1. Refer **Ingress Gateway** section in Grafana to determine increase in 4xx and 5xx error response codes. 2. Check **Ingress Gateway** logs on Kibana to determine the reason for the errors. For any additional guidance, contact My Oracle Support.

8.3.2.3 SM_INGRESS_ERROR_RATE_ABOVE_10_PERCENT

Table 8-119 SM_INGRESS_ERROR_RATE_ABOVE_10_PERCENT

Field	Details
Description	Transaction Error Rate detected above 10 Percent of Total on SM service (current value is: {{ $value }})
Summary	Transaction Error Rate detected above 10 Percent of Total Transactions
Severity	Critical
Condition	The number of failed transactions is above 10 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.36.1.2.4
Metric Used	ocpm_ingress_response_total
Recommended Actions	The alert gets cleared when the number of failed transactions are below 10% of the total transactions. To assess the reason for failed transactions, perform the following steps: 1. Check the service specific metrics to understand the service specific errors. For instance: `ocpm_ingress_response_total{servicename_3gpp="npcf-smpolicycontrol",response_code!~"2.*"}` 2. The service specific errors can be further filtered for errors specific to a method such as GET, PUT, POST, DELETE, and PATCH. For any additional guidance, contact My Oracle Support.

8.3.2.4 SM_EGRESS_ERROR_RATE_ABOVE_1_PERCENT

Table 8-120 SM_EGRESS_ERROR_RATE_ABOVE_1_PERCENT

Field	Details
Description	Egress Transaction Error Rate detected above 1 Percent of Total Transactions (current value is: {{ $value }})
Summary	Transaction Error Rate detected above 1 Percent of Total Transactions
Severity	Minor
Condition	The number of failed transactions is above 1 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.36.1.2.5
Metric Used	system_operational_state == 1
Recommended Actions	The alert gets cleared when the number of failed transactions are below 1% of the total transactions. To assess the reason for failed transactions, perform the following steps: 1. Check the service specific metrics to understand the service specific errors. For instance: `ocpm_egress_response_total{servicename_3gpp="npcf-smpolicycontrol",response_code!~"2.*"}` 2. The service specific errors can be further filtered for errors specific to a method such as GET, PUT, POST, DELETE, and PATCH. For any additional guidance, contact My Oracle Support.

8.3.2.5 PCF_CHF_INGRESS_TRAFFIC_RATE_ABOVE_THRESHOLD

Table 8-121 PCF_CHF_INGRESS_TRAFFIC_RATE_ABOVE_THRESHOLD

Field	Details
Description	User service Ingress traffic Rate from CHF is above threshold of Max MPS (current value is: {{ $value }})
Summary	Traffic Rate is above 90 Percent of Max requests per second
Severity	Major
Condition	The total User Service Ingress traffic rate from CHF has crossed the configured threshold of 900 TPS. Default value of this alert trigger point in PCF_Alertrules.yaml file is when user service Ingress Rate from CHF crosses 90% of maximum ingress requests per second.
OID	1.3.6.1.4.1.323.5.3.36.1.2.11
Metric Used	ocpm_userservice_inbound_count_total{service_resource="chf-service"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.2.6 PcfChfEgressErrorRateAbove10Percent

Table 8-122 PcfChfEgressErrorRateAbove10Percent

Field	Details
Description	Egress Transaction Error Rate detected above 10 Percent of Total on User service (current value is: {{ $value }})
Summary	Transaction Error Rate detected above 10 Percent of Total Transactions
Severity	Critical
Condition	The number of failed transactions from UDR is more than 10 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.36.1.2.12
Metric Used	ocpm_chf_tracking_response_total{servicename_3gpp="nchf-spendinglimitcontrol",response_code!~"2.*"}
Recommended Actions	The alert gets cleared when the number of failure transactions falls below the configured threshold. **Note**: Threshold levels can be configured using the `PCF_Alertrules.yaml` file. It is recommended to assess the reason for failed transactions. Perform the following steps to analyze the cause of increased traffic: 1. Refer **Egress Gateway** section in Grafana to determine increase in 4xx and 5xx error response codes. 2. Check **Egress Gateway** logs on Kibana to determine the reason for the errors. For any additional guidance, contact My Oracle Support.

8.3.2.7 PcfChfIngressErrorAboveMajorThreshold

Table 8-123 PcfChfIngressErrorAboveMajorThreshold

Field	Details
Description	Ingress Timeout Error Rate detected above 10 Percent of Total towards CHF service (current value is: {{ $value }})
Summary	Timeout Error Rate detected above 10 Percent of Total Transactions
Severity	Major
Condition	The number of failed transactions due to timeout is above 10 percent of the total transactions for CHF service.
OID	1.3.6.1.4.1.323.5.3.36.1.2.17
Metric Used	ocpm_chf_tracking_request_timeout_total{servicename_3gpp="nchf-spendinglimitcontrol"}
Recommended Actions	The alert gets cleared when the number of failed transactions due to timeout are below 10% of the total transactions. To assess the reason for failed transactions, perform the following steps: 1. Check the service specific metrics to understand the service specific errors. For instance: `ocpm_chf_tracking_request_timeout_total{servicename_3gpp="nchf-spendinglimitcontrol"}` 2. The service specific errors can be further filtered for errors specific to a method such as GET, PUT, POST, DELETE, and PATCH. For any additional guidance, contact My Oracle Support.

8.3.2.8 PCF_PENDING_BINDING_SITE_TAKEOVER

Table 8-124 PCF_PENDING_BINDING_SITE_TAKEOVER

Field	Details
Description	The site takeover configuration has been activated
Summary	The site takeover configuration has been activated
Severity	CRITICAL
Condition	sum by (application, container, namespace) (changes(occnp_pending_binding_site_takeover_total[2m])) > 0
OID	1.3.6.1.4.1.323.5.3.52.1.2.45
Metric Used	occnp_pending_binding_site_takeover_total
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.2.9 PCF_PENDING_BINDING_THRESHOLD_LIMIT_REACHED

Table 8-125 PCF_PENDING_BINDING_THRESHOLD_LIMIT_REACHED

Field	Details
Description	The Pending Operation table threshold has been reached.
Summary	The Pending Operation table threshold has been reached.
Severity	CRITICAL

Table 8-125 (Cont.) PCF_PENDING_BINDING_THRESHOLD_LIMIT_REACHED

Field	Details
Condition	sum by (application, container, namespace) (changes(occnp_threshold_limit_reached_total[2m])) > 0
OID	1.3.6.1.4.1.323.5.3.52.1.2.46
Metric Used	occnp_threshold_limit_reached_total
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.2.10 PCF_PENDING_BINDING_RECORDS_COUNT

Table 8-126 PCF_PENDING_BINDING_RECORDS_COUNT

Field	Details
Description	An attempt to internally recreate a PCF binding has been triggered by PCF
Summary	An attempt to internally recreate a PCF binding has been triggered by PCF
Severity	MINOR
Condition	sum by (application, container, namespace) (changes(occnp_pending_operation_records_count[10s])) > 0
OID	1.3.6.1.4.1.323.5.3.52.1.2.47
Metric Used	occnp_pending_operation_records_count
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.2.11 TDF_CONNECTION_DOWN

Table 8-127 TDF_CONNECTION_DOWN

Field	Details
Description	TDF connection is down.
Summary	TDF connection is down.
Severity	Critical
Condition	Diameter gateway raises an alert any time there is a disconnection with TDF peer node that is configured.
OID	1.3.6.1.4.1.323.5.3.52.1.2.48
Metric Used	occnp_diam_conn_app_network{applicationName="Sd"} == 0
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.2.12 AUTONOMOUS_SUBSCRIPTION_FAILURE

Table 8-128 AUTONOMOUS_SUBSCRIPTION_FAILURE

Field	Details
Description	Autonomous subscription failed for a configured Slice Load Level
Summary	Autonomous subscription failed for a configured Slice Load Level
Severity	Critical

Table 8-128 (Cont.) AUTONOMOUS_SUBSCRIPTION_FAILURE

Field	Details
Condition	The number of failed Autonomous Subscription for a configured Slice Load Leve in nwdaf-agent is greater than zero.
OID	1.3.6.1.4.1.323.5.3.52.1.2.49
Metric Used	subscription_failure{requestType="autonomous"}
Recommended Actions	The alert gets cleared when the failed Autonomous Subscription is corrected. To clear the alert, perform the following steps: 1. Delete the Slice Load Level configuration. 2. Re-provision the Slice Load Level configuration. For any additional guidance, contact My Oracle Support.

8.3.2.13 AM_NOTIFICATION_ERROR_RATE_ABOVE_1_PERCENT

Table 8-129 AM_NOTIFICATION_ERROR_RATE_ABOVE_1_PERCENT

Field	Details
Description	AM Notification Error Rate detected above 1 Percent of Total (current value is: {{ $value }})
Summary	AM Notification Error Rate detected above 1 Percent of Total (current value is: {{ $value }})
Severity	MINOR
Condition	(sum(rate(http_out_conn_response_total{pod=~".*amservice.*",responseCode!~"2.*",servicename3gpp="npcf-am-policy-control"}[1d])) / sum(rate(http_out_conn_response_total{pod=~".*amservice.*",servicename3gpp="npcf-am-policy-control"}[1d]))) * 100 >= 1
OID	1.3.6.1.4.1.323.5.3.52.1.2.54
Metric Used	http_out_conn_response_total
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.2.14 AM_AR_ERROR_RATE_ABOVE_1_PERCENT

Table 8-130 AM_AR_ERROR_RATE_ABOVE_1_PERCENT

Field	Details
Description	Alternate Routing Error Rate detected above 1 Percent of Total on AM Service (current value is: {{ $value }})
Summary	Alternate Routing Error Rate detected above 1 Percent of Total on AM Service (current value is: {{ $value }})
Severity	MINOR
Condition	(sum by (fqdn) (rate(ocpm_ar_response_total{pod=~".*amservice.*",responseCode!~"2.*",servicename3gpp="npcf-am-policy-control"}[1d])) / sum by (fqdn) (rate(ocpm_ar_response_total{pod=~".*amservice.*",servicename3gpp="npcf-am-policy-control"}[1d]))) * 100 >= 1
OID	1.3.6.1.4.1.323.5.3.52.1.2.55
Metric Used	ocpm_ar_response_total

Table 8-130 (Cont.) AM_AR_ERROR_RATE_ABOVE_1_PERCENT

Field	Details
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.2.15 UE_NOTIFICATION_ERROR_RATE_ABOVE_1_PERCENT

Table 8-131 UE_NOTIFICATION_ERROR_RATE_ABOVE_1_PERCENT

Field	Details
Description	UE Notification Error Rate detected above 1 Percent of Total (current value is: {{ $value }})
Summary	UE Notification Error Rate detected above 1 Percent of Total (current value is: {{ $value }})
Severity	MINOR
Condition	(sum(rate(http_out_conn_response_total{pod=~".*ueservice.*",responseCode!~"2.*",servicename3gpp="npcf-ue-policy-control"}[1d])) / sum(rate(http_out_conn_response_total{pod=~".*ueservice.*",servicename3gpp="npcf-ue-policy-control"}[1d]))) * 100 >= 1
OID	1.3.6.1.4.1.323.5.3.52.1.2.56
Metric Used	http_out_conn_response_total
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.2.16 UE_AR_ERROR_RATE_ABOVE_1_PERCENT

Table 8-132 UE_AR_ERROR_RATE_ABOVE_1_PERCENT

Field	Details
Description	Alternate Routing Error Rate detected above 1 Percent of Total on UE Service (current value is: {{ $value }})
Summary	Alternate Routing Error Rate detected above 1 Percent of Total on UE Service (current value is: {{ $value }})
Severity	MINOR
Condition	(sum by (fqdn) (rate(ocpm_ar_response_total{pod=~".*ueservice.*",responseCode!~"2.*",servicename3gpp="npcf-ue-policy-control"}[1d])) / sum by (fqdn) (rate(ocpm_ar_response_total{pod=~".*ueservice.*",servicename3gpp="npcf-ue-policy-control"}[1d]))) * 100 >= 1
OID	1.3.6.1.4.1.323.5.3.52.1.2.57
Metric Used	ocpm_ar_response_total
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.3 PCRF Alerts

This section provides information about PCRF alerts.

8.3.3.1 PRE_UNREACHABLE_EXCEEDS_CRITICAL_THRESHOLD

PRE_UNREACHABLE_EXCEEDS_CRITICAL_THRESHOLD

Table 8-133 PRE_UNREACHABLE_EXCEEDS_CRITICAL_THRESHOLD

Field	Details
Description	PRE fail count exceeds the critical threshold limit.
Summary	Alert PRE unreachable NS:{{ $labels.kubernetes_namespace }}, PODNAME: {{ $labels.kubernetes_pod_name }}, INST:{{ $labels.instance }} REL: {{ $labels.release }}
Severity	Critical
Condition	PRE fail count exceeds the critical threshold limit.
OID	1.3.6.1.4.1.323.5.3.44.1.2.9
Metric Used	http_out_conn_response_total{container="pcrf-core", responseCode!~"2.*", serviceResource="PRE"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.3.2 PcrfDown

Table 8-134 PcrfDown

Field	Details
Description	PCRF Service is down
Summary	Alert PCRF_DOWN NS:{{ $labels.kubernetes_namespace }}, PODNAME: {{ $labels.kubernetes_pod_name }}, INST:{{ $labels.instance }} REL: {{ $labels.release }}
Severity	Critical
Condition	None of the pods of the PCRF service are available.
OID	1.3.6.1.4.1.323.5.3.44.1.2.33
Metric Used	appinfo_service_running{service=~".*pcrf-core"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.3.3 CCA_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

CCA_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Table 8-135 CCA_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Field	Details
Description	CCA fail count exceeds the critical threshold limit
Summary	Alert CCA_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD NS: {{ $labels.kubernetes_namespace }}, PODNAME:{{ $labels.kubernetes_pod_name }}, INST:{{ $labels.instance }} REL:{{ $labels.release }}
Severity	Critical
Condition	The failure rate of CCA messages has exceeded the configured threshold limit.
OID	1.3.6.1.4.1.323.5.3.44.1.2.13

Table 8-135 (Cont.) CCA_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Field	Details
Metric Used	occnp_diam_response_local_total{msgType=~"CCA.*", responseCode!~"2.*"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.3.4 AAA_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

AAA_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Table 8-136 AAA_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Field	Details
Description	AAA fail count exceeds the critical threshold limit
Summary	Alert AAA_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD NS: {{ $labels.kubernetes_namespace }}, PODNAME:{{ $labels.kubernetes_pod_name }}, INST:{{ $labels.instance }} REL:{{ $labels.release }}
Severity	Critical
Condition	The failure rate of AAA messages has exceeded the critical threshold limit.
OID	1.3.6.1.4.1.323.5.3.36.1.2.34
Metric Used	occnp_diam_response_local_total{msgType=~"AAA.*", responseCode!~"2.*"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.3.5 RAA_RX_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

RAA_RX_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Table 8-137 RAA_RX_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Field	Details
Description	RAA Rx fail count exceeds the critical threshold limit
Summary	Alert RAA_Rx_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD NS: {{ $labels.kubernetes_namespace }}, PODNAME:{{ $labels.kubernetes_pod_name }}, INST:{{ $labels.instance }} REL:{{ $labels.release }}
Severity	Critical
Condition	The failure rate of RAA Rx messages has exceeded the configured threshold limit.
OID	1.3.6.1.4.1.323.5.3.36.1.2.35
Metric Used	occnp_diam_response_local_total{msgType="RAA", appType="Rx", responseCode!~"2.*"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.3.6 RAA_GX_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

RAA_GX_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Table 8-138 RAA_GX_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Field	Details
Description	RAA Gx fail count exceeds the critical threshold limit
Summary	Alert RAA_GX_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD NS: {{ $labels.kubernetes_namespace }}, PODNAME:{{ $labels.kubernetes_pod_name }}, INST:{{ $labels.instance }} REL:{{ $labels.release }}
Severity	Critical
Condition	The failure rate of RAA Gx messages has exceeded the configured threshold limit.
OID	1.3.6.1.4.1.323.5.3.44.1.2.18
Metric Used	occnp_diam_response_local_total{msgType="RAA", appType="Gx", responseCode!~"2.*"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.3.7 ASA_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

ASA_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Table 8-139 ASA_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Field	Details
Description	ASA fail count exceeds the critical threshold limit
Summary	Alert ASA_FAIL_COUNT_EXCEEDS_CRITICAL_THRESHOLD NS: {{ $labels.kubernetes_namespace }}, PODNAME:{{ $labels.kubernetes_pod_name }}, INST:{{ $labels.instance }} REL:{{ $labels.release }}
Severity	Critical
Condition	The failure rate of ASA messages has exceeded the configured threshold limit.
OID	1.3.6.1.4.1.323.5.3.44.1.2.17
Metric Used	occnp_diam_response_local_total{msgType=~"ASA.*", responseCode!~"2.*"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.3.8 ASATimeoutlCountExceedsThreshold

ASA_TIMEOUT_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Table 8-140 ASA_TIMEOUT_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Field	Details
Description	ASA timeout count exceeds the critical threshold limit
Summary	Alert ASA_TIMEOUT_COUNT_EXCEEDS_CRITICAL_THRESHOLD NS: {{ $labels.kubernetes_namespace }}, PODNAME:{{ $labels.kubernetes_pod_name }}, INST:{{ $labels.instance }} REL:{{ $labels.release }}
Severity	Critical
Condition	The timeout rate of ASA messages has exceeded the configured threshold limit.

Table 8-140 (Cont.) ASA_TIMEOUT_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Field	Details
OID	1.3.6.1.4.1.323.5.3.44.1.2.31
Metric Used	occnp_diam_response_local_total{msgType="ASA", responseCode="timeout"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.3.9 RAA_RX_TIMEOUT_COUNT_EXCEEDS_CRITICAL_THRESHOLD

RAA Rx Timeout Count Exceeds Critical Threshold

Table 8-141 RAA Rx Timeout Count Exceeds Critical Threshold

Field	Details
Description	RAA Rx timeout count exceeds the critical threshold limit
Summary	Alert RAA_RX_TIMEOUT_COUNT_EXCEEDS_CRITICAL_THRESHOLD NS: {{ $labels.kubernetes_namespace }}, PODNAME:{{ $labels.kubernetes_pod_name }}, INST:{{ $labels.instance }} REL:{{ $labels.release }}
Severity	Critical
Condition	The timeout rate of RAA Rx messages has exceeded the configured threshold limit.
OID	1.3.6.1.4.1.323.5.3.36.1.2.36
Metric Used	occnp_diam_response_local_total{msgType="RAA", appType="Rx", responseCode!~"timeout"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.3.10 RAA_GX_TIMEOUT_COUNT_EXCEEDS_CRITICAL_THRESHOLD

RAA_GX_TIMEOUT_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Table 8-142 RAA_GX_TIMEOUT_COUNT_EXCEEDS_CRITICAL_THRESHOLD

Field	Details
Description	RAA Gx timeout count exceeds the critical threshold limit
Summary	Alert RAA_GX_TIMEOUT_COUNT_EXCEEDS_CRITICAL_THRESHOLD NS: {{ $labels.kubernetes_namespace }}, PODNAME:{{ $labels.kubernetes_pod_name }}, INST:{{ $labels.instance }} REL:{{ $labels.release }}
Severity	Critical
Condition	The timeout rate of RAA Gx messages has exceeded the configured threshold limit.
OID	1.3.6.1.4.1.323.5.3.44.1.2.32
Metric Used	occnp_diam_response_local_total{msgType="RAA", appType="Gx", responseCode!~"timeout"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.3.11 RESPONSE_ERROR_RATE_ABOVE_CRITICAL_PERCENT

RESPONSE_ERROR_RATE_ABOVE_CRITICAL_PERCENT

Table 8-143 RESPONSE_ERROR_RATE_ABOVE_CRITICAL_PERCENT

Field	Details
Description	CCA, AAA, RAA, ASA and STA error rate combined is above 10 percent
Summary	Alert RESPONSE_ERROR_RATE_ABOVE_CRITICAL_PERCENT NS: {{ $labels.kubernetes_namespace }}, PODNAME:{{ $labels.kubernetes_pod_name }}, INST:{{ $labels.instance }} REL:{{ $labels.release }}
Severity	Critical
Condition	The combined failure rate of CCA, AAA, RAA, ASA, and STA messages is more than 10% of the total responses.
OID	1.3.6.1.4.1.323.5.3.36.1.2.37
Metric Used	occnp_diam_response_local_total{ responseCode!~"2.*"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.3.12 Rx_RESPONSE_ERROR_RATE_ABOVE_CRITICAL_PERCENT

Rx_RESPONSE_ERROR_RATE_ABOVE_CRITICAL_PERCENT

Table 8-144 Rx_RESPONSE_ERROR_RATE_ABOVE_CRITICAL_PERCENT

Field	Details
Description	Rx error rate combined is above 10 percent
Summary	Alert Rx_RESPONSE_ERROR_RATE_ABOVE_CRITICAL_PERCENT NS: {{ $labels.kubernetes_namespace }}, PODNAME:{{ $labels.kubernetes_pod_name }}, INST:{{ $labels.instance }} REL:{{ $labels.release }}
Severity	Critical
Condition	The failure rate of Rx responses is more than 10% of the total responses.
OID	1.3.6.1.4.1.323.5.3.36.1.2.38
Metric Used	occnp_diam_response_local_total{ responseCode!~"2.*", appType="Rx"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

8.3.3.13 Gx_RESPONSE_ERROR_RATE_ABOVE_CRITICAL_PERCENT

Gx_RESPONSE_ERROR_RATE_ABOVE_CRITICAL_PERCENT

Table 8-145 Gx_RESPONSE_ERROR_RATE_ABOVE_CRITICAL_PERCENT

Field	Details
Description	Gx error rate combined is above 10 percent
Summary	Alert Gx_RESPONSE_ERROR_RATE_ABOVE_CRITICAL_PERCENT NS: {{ $labels.kubernetes_namespace }}, PODNAME:{{ $labels.kubernetes_pod_name }}, INST:{{ $labels.instance }} REL:{{ $labels.release }}
Severity	Critical
Condition	The failure rate of Gx responses is more than 10% of the total responses.

Table 8-145 (Cont.) Gx_RESPONSE_ERROR_RATE_ABOVE_CRITICAL_PERCENT

Field	Details
OID	1.3.6.1.4.1.323.5.3.36.1.2.39
Metric Used	occnp_diam_response_local_total{ responseCode!~"2.*", appType="Gx"}
Recommended Actions	For any additional guidance, contact My Oracle Support.

9
CNC Policy Metrics

This chapter includes information about Metrics for Oracle Communications Cloud Native Core, Policy (CNC Policy).

Policy uses the observability tool Grafana to analyze and visualize data. The Grafana Dashboard consists of panels displaying the data as graphs, charts or other visualizations. A dashboard in Grafana is represented by a JSON object, which stores the metadata of its dashboard. Following Grafana dashboards are created for Policy Observability and debugging purposes:

- **Observability Dashboard**:
  - This dashboard metadata is stored in the `Policy_Observability_Dashboard.json` file.
  - This dashboard has the panels that monitors the overall health status of the setup & contains details like resource utilization, kmps, latency, etc.
  - Customers can use this Dashboard on a regular basis for observing the status of the setup and also for assessing the setup performance with a run.
  - Depending upon the applicability of a panel or row, the customers have the flexibility to either remove, update or add content to the Dashboard.
- **Debug Dashboard**:
  - This dashboard metadata is stored in the `Policy_Debug_Dashboard.json` file.
  - This dashboard has the panels that monitor the operational status of the setup and is used for debugging & troubleshooting purposes.
  - Customers can use this Dashboard when any issue occur in the NF and also when reporting a probable issue to Oracle Customer Support from the setup.
  - Depending upon the applicability of a panel or row, the customers have the flexibility to either remove, update or add content to the Dashboard.

The name of the metrics may contain suffix such as total, seconds, max and so on. It gets added by the Micrometer registry if it is not present in the metrics name. The metric name has the following format for suffix:

The metric name is equal to <Basename of the metric>_<Suffix>

Table 9-1 Metrics type and Suffix

Metric Type	Suffix
Counter	_total
Gauge	N/A
TimerGauge	_seconds
MultiGauge	N/A

Table 9-1 (Cont.) Metrics type and Suffix

Metric Type	Suffix
Timer	_seconds_max or _seconds **Note**: There are two types of suffix used for timer metrics. For example, my_timer_seconds_max gauge and my_timer_seconds summary. In summary type, there will be further addition of suffix such as _count or _sum.
DistributionSummary	N/A or _max **Note**: There are two types of suffix used for DistributionSummary. For example, my_distribution_ratio histogram and my_distribution_ratio_max gauge. In the histogram type there will be further addition of suffix such as bucket, _count, or sum.
LongTaskTimer	_seconds_max or _seconds **Note:** There are two types of suffix used for LongTaskTimer. For example, long_task_timer_seconds_max gauge and long_task_timer_seconds summary. In summary type there will be further addition of suffix such as _active_count or _duration_sum.

Table 9-2 Dimension Description

Dimension	Description
operation_type	Type of operation Values: • create • get • put • update • terminate • update_notify • terminate_notify • subscribe • unsubscribe • transfer • resubscribe
dnn	Data Network Name or Access Point Name
snssai	Single Network Slice Selection Assistance Information
response_code	Response code **HTTP interfaces:** • 1xx • 2xx • 3xx • 4xx • 5xx **Diameter interfaces:** • 2xxx • 3xxx • 4xxx • 5xxx

Table 9-2 (Cont.) Dimension Description

Dimension	Description
latency	The total time in between request and response. If latency between request and response is 203, then bucket number is 4. Max bucket set to 10 (0-9), Range 50ms.
nf_instance_Id	Unique id of the nf Instance. • ingress: source nfInstanceId • egress: destination nfInstanceId **HTTP interfaces:** **Diameter interfaces:** • ingress: Origin-Host AVP • egress: Destination-Host AVP
nf_name	This represents the FQDN corresponding to the NF InstanceID present in the nf_instance_id dimension. **HTTP interface:** egress: NF FQDN
sbi_priority	Service Based Interface
service_version	Service version Value: [UDR = "v1,v2", CHF = "v1"]
service	The complete name of current service. Value: string
namespace	The namespace of current service. Value: string
category	The category of current service. Value: • database • common • infra • pcf • bsf
destHost	Value of destination Host received or sent in the corresponding request message
destRealm	Value of destination Realm received or sent in the corresponding request message
origHost	Value of origination Host received or sent in the corresponding request message
origRealm	Value of origination Realm received or sent in the corresponding request message
reqDestHost	Value of destination Host in corresponding request message of response message.
reqDestRealm	Value of destination Realm in corresponding request message of response message.
reqOrigHost	Value of origination Host in corresponding request message of response message.
reqOrigRealm	Value of origination Realm in corresponding request message of response message.
direction	Indicates direction of message flow. • "In" means coming towards POD/micro-service • "Out" means going out from POD/micro-service
appId	Application ID exchanged in CEX messages or used in the respective message of an application.

Table 9-2 (Cont.) Dimension Description

Dimension	Description
applicationName	Human readable name of corresponding application ID 16777236 => Rx 16777238 => Gx 16777302 => Sy 16777217 => Sh **Note**: Sh interface is not supported for Converged Policy mode of deployment. 0xffffffff => Relay
cmdCode	Command code value in the received or sent, request or answer message
msgType	Type of the message, for example CCRT-T, CCR-I etc.
responseCode	result code in diameter message
spendingLimitDataSource	Specifies the source from which PCF fetches policy counters. Value: OCS
retry	Identify message is a retry message. Value: • true • false
retryAnswer	Reason for the retry message. Value: • error code • timeout
level	Indicates the current load level or the level of pod congestion. Value: • 0 = Normal • 1 = DOC • 2 = Congested
type	Resource type Value: • PendingRequest • CPU • Memory
le	le is abrreviated as "Less than equal to".Value of a defined bucket for a Histogram.
sessRuleReports	Indicates that session rule report is received at PCF.
policyDecFailureReports	Indicates that Policy decision failure report is received at PCF.
isLeaderPod	Indicates if the pod calculating the threshold level is a leader pod.
prevLevel	Indicates the previous load level prior to current load level calculation.
levelChangeType	Indicates the level change type. The value of this dimension can be: • None: when load level is same • Increment: when level changes to higher level
servicenameNon3gpp	
serviceResource	
params	Lists the API parameters.
outcome	Shows the outcome of an operation such as SUCCESS, FAILURE, TIMEOUT.
cause	Contains the error cause.

9.1 Undertow Server Metrics

This section describes the metrics and examples for Undertow Server Metrics.

Table 9-3 undertow_queue_limiter_reject_request

Field	Details
Description	Counter metrics to track the total number of requests rejected by undertow queue request limiter. It includes "message_priority" which denotes the message priority of messages that are rejected.
Type	Counter
Dimension	• application • message_priority
Examples	• occnp_undertow_queue_limiter_reject_request_total{application="policyds",message_priority="31",} 26.0

Table 9-4 undertow_queue_limiter_accept_request

Field	Details
Description	Counter metrics to track the total number of requests accepted by undertow queue request limiter. It includes "message_priority" which denotes the message priority of messages that are accepted.
Type	Counter
Dimension	• application • message_priority
Example	• occnp_undertow_queue_limiter_accept_request_total{application="policyds",message_priority="5",} 100.0

Table 9-5 undertow_queue_request_limiter_active_threads_count

Field	Details
Description	This displays number of active threads of threadpool that executes the undertow queue request limiter.
Type	Gauge
Dimension	• application
Example	• occnp_undertow_queue_request_limiter_active_threads_count{application="policyds",} 1.0

9.2 TLS Metrics

The following table describes the TLS metrics and the respective dimensions:

Table 9-6 oc_ingressgateway_incoming_tls_connections

Field	Details
Description	Number of TLS Connections received on the Ingress Gateway and their negotiated TLS versions. The versions can be TLSv1.3 or TLSv1.2
Type	Gauge
Dimensions	• host • NegotiatedTLSVersion • direction • instanceIdentifier
Example	-

Table 9-7 oc_egressgateway_incoming_tls_connections

Field	Details
Description	Number of TLS Connections received on the Egress Gateway and their negotiated TLS versions. The versions can be TLSv1.3 or TLSv1.2
Type	Gauge
Dimensions	• host • NegotiatedTLSVersion • direction • instanceIdentifier
Example	-

Table 9-8 security_cert_x509_expiration_seconds

Field	Details
Description	Indicates the time to certificate expiry in epoch seconds.
Type	Histogram
Dimensions	serialNumber
Example	-

Table 9-9 diam_conn_network

Field	Details
Description	Indicates the number of TLS connections per TLS version.
Type	Gauge
Dimensions	• tlsversion • peerHost • peerRealm
Example	-

Table 9-10 diam_failed_conn_network

Field	Details
Description	Indicates the number of failed TLS connections. **Note:** It is applicable when we configure **Initiate Connection** to `true` in peer node configurations in the CNC Console.
Type	Gauge
Dimensions	• tlsversion • reason
Example	diam_failed_conn_network{peerName="dgw",reason="SSL Handshake Exception",} 1.0

Table 9-11 diam_conn_network_responder

Field	Details
Description	Indicates the number of allowed TLS responder connections with or without the peer configuration.
Type	Gauge
Dimensions	• tlsversion • peerconfigvalidated • peerHost • peerRealm
Example	-

Table 9-12 dgw_tls_cert_expiration_seconds

Field	Details
Description	Indicates the number of allowed TLS responder connections with or without the peer configuration.
Type	Gauge
Dimensions	• serialNumber (a number assigned by CA to each certificate) • subject (information about the cert issuer)
Example	dgw_tls_cert_expiration_seconds{serialNumber="122859034516052844067924392230910496431389299229",subject="OU=CGIU, O=ORCL, L=BLR, ST=KA, C=IN",} 1.797882352E9

9.3 Egress Gateway Metrics for SCP

> **Note**
>
> Peer health pings happen in Primary pod only and not in secondary pod due to which metrics are getting pegged in Primary pod only.

This section provides details about SCP health monitoring metrics and the respective dimensions.

Table 9-13 oc_egressgateway_peer_health_status

Field	Details
Description	It defines Egress Gateway peer health status. • This metric is set to 1, if a peer is unhealthy. • This metric is reset to 0, when it becomes healthy again. • This metric is set to -1, if peer is removed from peerconfiguration.
Metric Type	Gauge
Dimensions	• peer • vfqdn

Examples:

- oc_egressgateway_peer_health_status{"peer":"10.75.213.172:8080"} 1.0
- oc_egressgateway_peer_health_status{"peer":"10.75.213.172:8080"} 0.0
- oc_egressgateway_peer_health_status{"vfqdn":"http://abc.com","peer":"10.75.213.172:8080"} 1.0
- oc_egressgateway_peer_health_status{"vfqdn":"http://abc.com","peer":"10.75.213.172:8080"} 0.0

Table 9-14 oc_egressgateway_peer_health_ping_request_total

Field	Details
Description	This metric is incremented every time a health ping is sent toward a peer.
Metric Type	Counter
Dimensions	• peer • vfqdn

Examples:

- oc_egressgateway_peer_health_ping_request_total{"peer":"10.75.213.172:8080"} 389.0
- oc_egressgateway_peer_health_ping_request_total{"peer":"10.75.213.172:8080"} 439.0
- oc_egressgateway_peer_health_ping_request_total{"vfqdn":"http://abc.com","peer":"10.75.213.172:8080"} 389.0
- oc_egressgateway_peer_health_ping_request_total{"vfqdn":"http://abc.com","peer":"10.75.213.172:8080"} 439.0

Table 9-15 oc_egressgateway_peer_health_ping_response_total

Field	Details
Description	This metric is incremented every time a health ping response is received from a peer irrespective of success or failure.
Metric Type	Counter
Dimensions	• peer • vfqdn • statusCode • cause

Examples:

- oc_egressgateway_peer_health_ping_response_total{"peer":"10.75.213.172:8080","status":"httpstatus","cause":""} 89.0
- oc_egressgateway_peer_health_ping_response_total{"peer":"10.75.213.172:8080","status":"Exception","cause":"exception cause"} 39.0
- oc_egressgateway_peer_health_ping_response_total{"vfqdn":"http://abc.com","status":"httpstatus","cause":""} 89.0
- oc_egressgateway_peer_health_ping_response_total{"vfqdn":"http://abc.com","status":"Exception","cause":"exception cause"} 39.0

Table 9-16 oc_egressgateway_peer_health_status_transitions_total

Field	Details
Description	This metric is incremented every time a peer is transitioned from Availble to Unavailable or from Unavailable to Available.
Metric Type	Counter
Dimensions	• peer • vfqdn • from • to

Examples:

- oc_egressgateway_peer_health_status_transitions_total{"identifier":"10.75.213.172:8080","from":"available","to":"unavailable"} 14.0
- oc_egressgateway_peer_health_status_transitions_total{"identifier":"10.75.213.172:8080","from":"unavailable","to":"available"} 34.0
- oc_egressgateway_peer_health_status_transitions_total{"vfqdn":"http://abc.com",*"peer":"10.75.213.172:8080",*"from":"unavailable","to":"available"} 34.0
- oc_egressgateway_peer_health_status_transitions_total{"vfqdn":"http://abc.com","peer":"10.75.213.172:8080","from":"available","to":"unavailable"} 14.0

Table 9-17 oc_egressgateway_peer_count

Field	Details
Description	This metric is incremented every time for the peer count.
Metric Type	Gauge
Dimensions	• peerset

Examples:

- oc_egressgateway_peer_count{"peerset":"set-0"} 3.0

Table 9-18 oc_egressgateway_peer_available_count

Field	Details
Description	This metric is incremented every time for the available peer count.

Table 9-18 (Cont.) oc_egressgateway_peer_available_count

Field	Details
Metric Type	Gauge
Dimensions	• peerset

Examples:

- oc_egressgateway_peer_available_count{"peerset":"set-0"} 4.0

9.4 Correlation-Info Header Metrics

For every correlation-info header received or newly generated, a metric is pegged. Following are the list of metrics:

Below table lists the metrics that is implemented as part of this feature with the following Dimensions:

1. operation_type= {"create","update","delete","subscribe","unsubscribe","terminate","register","deregister"}
2. correlation_info_type={"imsi", "msisdn", "imsi,msisdn"}

Table 9-19 occnp_correlation_info_header_received

Field	Details
Description	PCF reports the total incoming requests that are carrying the correlation-info header.
Metric Type	Counter
Dimensions	• operation_type • correlation_info_type

Example:

- occnp_correlation_info_header_received_total '{correlation_info_type="imsi", operation_type="create"}',2
- occnp_correlation_info_header_received_total '{correlation_info_type="imsi", operation_type="update"}',1

Table 9-20 occnp_correlation_info_header_forwarded

Field	Details
Description	PCF reports the total outgoing requests that are carrying the correlation-info header.
Metric Type	Counter
Dimensions	• operation_type • correlation_info_type

Example:

- occnp_correlation_info_header_forwarded_total '{correlation_info_type="imsi", operation_type="subscribe"}',2

Table 9-21 occnp_correlation_info_header_generated

Field	Details
Description	PCF reports the total responses that are carrying the correlation-info header.
Metric Type	Counter
Dimensions	• operation_type • correlation_info_type

Example:

- occnp_correlation_info_header_generated_total '{correlation_info_type="imsi", operation_type="create"}',2

9.5 Config Server Metrics

The following table describes the Config Server metrics and respective dimensions:

Table 9-22 topic_version

Field	Details
Description	Config-service will have this metrics a database value from each topic version. The Services fetching the configurations from Configuration Server, will have its current topic version till which configurations has been fetched successfully. **Note**: This counter metric keep updating on every successful iteration.
Metric Type	Gauge
Dimensions	• Service Name • Pod Name

Examples:

- topic_version{topicName="pcf.public.sessionrule",} 77.0
- topic_version{topicName="common.logging.diam-connector",} 53.0
- topic_version{topicName="pcf.userservice.cfg",} 35.0
- topic_version{topicName="pcrf.public.networkelement.pgw",} 2.0
- topic_version{topicName="NRF.BSF",} 20.0
- topic_version{topicName="common.bindingservice.cfg",} 6.0
- topic_version{topicName="common.logging.policy-ds",} 21.0
- topic_version{topicName="NRF.UDR",} 22.0
- topic_version{topicName="common.logging.diam-gateway",} 93.0
- topic_version{topicName="common.logging.pcf-user",} 35.0
- topic_version{topicName="common.logging.pcf-sm",} 78.0
- topic_version{topicName="pcrf.coreservice",} 3.0

- topic_version{topicName="public.policy.project",} 148.0
- topic_version{topicName="Subscriptions",} 14.0
- topic_version{topicName="pcf.smservice.cfg",} 126.0
- topic_version{topicName="common.public.feature1",} 10.0
- topic_version{topicName="common.public.diampeernode",} 2.0
- topic_version{topicName="common.logging.pcrf-core",} 2.0
- topic_version{topicName="pcf.public.authorizeddefaultqos",} 77.0
- topic_version{topicName="public.policy.project.content",} 85.0
- topic_version{topicName="pcf.public.pccrule",} 85.0
- topic_version{topicName="NRF.CHF",} 52.0
- topic_version{topicName="config.level",} 1053.0
- topic_version{topicName="pds.public.settings",} 2.0

Table 9-23 occnp_config_server_overall_processing_time_seconds

Field	Details
Description	Configuration server's overall processing time.
Type	Summary
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Table 9-24 occnp_config_server_overall_processing_time_seconds_max

Field	Details
Description	Configuration service overall processing time.
Type	Gauge
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Examples:

- occnp_config_server_overall_processing_time_seconds_max{error="none", exception="none", method="POST", outcome="SUCCESS", status="201", uri="/binding/v1/contextBinding/context-owner/{contextOwner}",} 2.595594434

- occnp_config_server_overall_processing_time_seconds_max{error="none", exception="none", method="DELETE", outcome="SUCCESS", status="204", uri="/binding/v1/contextBinding/contextId/{contextId}",} 0.276239293
- occnp_config_server_overall_processing_time_seconds_max{error="ContextBindingNotFound", exception="ContextBindingNotFound", method="DELETE", outcome="CLIENT_ERROR", status="404", uri="/binding/v1/contextBinding/cleanup",} 0.115288423

Table 9-25 occnp_config_server_overall_processing_time_seconds_count

Field	Details
Description	Configuration service overall processing time.
Type	Counter
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Examples:

- occnp_config_server_overall_processing_time_seconds_count{error="none", exception="none", method="POST", outcome="SUCCESS", status="201", uri="/binding/v1/contextBinding/context-owner/{contextOwner}",} 3.0
- occnp_config_server_overall_processing_time_seconds_count{error="none", exception="none", method="DELETE", outcome="SUCCESS", status="204", uri="/binding/v1/contextBinding/contextId/{contextId}",} 3.0
- occnp_config_server_overall_processing_time_seconds_count{error="ContextBindingNotFound", exception="ContextBindingNotFound",method="DELETE",outcome="CLIENT_ERROR",status="404",uri="/binding/v1/contextBinding/cleanup",} 6.0

Table 9-26 occnp_config_server_overall_processing_time_seconds_sum

Field	Details
Description	Configuration service overall processing time.
Type	Counter
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Examples:

- occnp_config_server_overall_processing_time_seconds_sum{error="none",exception="none",method="POST", outcome="SUCCESS", status="201", uri="/binding/v1/contextBinding/context-owner/{contextOwner}",} 5.861114721
- occnp_config_server_overall_processing_time_seconds_sum{error="none", exception="none",method="DELETE",outcome="SUCCESS",status="204",uri="/binding/v1/contextBinding/contextId/{contextId}",} 0.349598291
- occnp_config_server_overall_processing_time_seconds_sum{error="ContextBindingNotFound",exception="ContextBindingNotFound", method="DELETE",outcome="CLIENT_ERROR",status="404", uri="/binding/v1/contextBinding/cleanup",} 0.203103476

Table 9-27 occnp_config_server_db_operation_time_seconds_count

Field	Details
Description	Configuration service overall database processing time.
Type	Gauge
Dimension	• exception • method • repository • state

Examples:

- occnp_config_server_db_operation_time_seconds_count{exception="None",method="findByTopicInfo",repository="ConfigurationItemRepository",state="SUCCESS",} 7721880.0
- occnp_config_server_db_operation_time_seconds_count{exception="None",method="findByName",repository="TopicInfoRepository",state="SUCCESS",} 8190322.0

Table 9-28 occnp_config_server_db_operation_time_seconds_sum

Field	Details
Description	Configuration service overall database processing time.
Type	Gauge
Dimension	• exception • method • repository • state

Examples:

- occnp_config_server_db_operation_time_seconds_sum{exception="None",method="findByTopicInfo",repository="ConfigurationItemRepository",state="SUCCESS",} 4713.01
- occnp_config_server_db_operation_time_seconds_sum{exception="None",method="findByName",repository="TopicInfoRepository",state="SUCCESS",} 6328.74490

Table 9-29 occnp_config_server_db_operation_time_seconds_max

Field	Details
Description	Configuration service overall database processing time.
Type	Gauge

Table 9-29 (Cont.) occnp_config_server_db_operation_time_seconds_max

Field	Details
Dimension	• exception • method • repository • state

9.6 SM Service Metrics

The following table describes the SM Service metrics and respective dimensions:

Table 9-30 ocpm_ingress_request_total

Field	Details
Description	Total number of ingress requests received.
Metric Type	Counter
Dimensions	• operation_type • dnn • snssai • nf_instance_id • sbi_priority • servicename_3gpp • application • hostname

Examples:

- ocpm_ingress_request_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="",operation_type="create", sbi_priority="",servicename_3gpp="npcf-smpolicycontrol",snssai="11-abc123",} 1.0 ; Type-Counter
- ocpm_ingress_request_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="",operation_type="update",sbi_priority="",servicename_3gpp="npcf-smpolicycontrol",snssai="11-abc123",} 1.0 ; Type-Counter
- ocpm_ingress_request_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="",operation_type="delete",sbi_priority="",servicename_3gpp="npcf-smpolicycontrol,snssai="11-abc123",} 1.0 ; Type-Counter
- ocpm_ingress_request_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="",operation_type="get",sbi_priority="",servicename_3gpp="npcf-smpolicycontrol",snssai="11-abc123",} 1.0 ; Type-Counter

Table 9-31 ocpm_ingress_response_total

Field	Details
Description	Total number of ingress responses.
Metric Type	Counter

Table 9-31 (Cont.) ocpm_ingress_response_total

Field	Details
Dimensions	• operation_type • dnn • snssai • nf_instance_id • sbi_priority • servicename_3gpp • response_code • binding_level • binding_id • application • hostname • spendingLimitDataSource • cause

Examples:

- ocpm_ingress_response_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="", operation_type="create",response_code="2xx",sbi_priority="",servicename_3gpp="npcf-smpolicycontrol",snssai="11-abc123",} 1.0 ; Type-Counter
- ocpm_ingress_response_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="", operation_type="update",response_code="4xx",sbi_priority="",servicename_3gpp="npcf-smpolicycontrol",snssai="11-abc123",binding_level="NF_SET(1)", binding_id="setxyz.pcfset.5gc.mnc015.mcc345"} 1.0 ; Type-Counter
- ocpm_ingress_response_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="", operation_type="delete",response_code="2xx",sbi_priority="",servicename_3gpp="npcf-smpolicycontrol",snssai="11-abc123",binding_level="NF_SET(1)", binding_id="setxyz.pcfset.5gc.mnc015.mcc345"} 1.0 ; Type-Counter
- ocpm_ingress_response_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="", operation_type="get",response_code="2xx",sbi_priority="",servicename_3gpp="npcf-smpolicycontrol",snssai="11-abc123",binding_level="NF_SET(1)", binding_id="setxyz.pcfset.5gc.mnc015.mcc345"} 1.0 ; Type-Counter
- ocpm_ingress_response_total{application="pcf_smservice",binding_id="",binding_level="", cause="PENDING_TRANSACTION",dnn="dnn1",hostname="jiandong-pcf-smservice-6dbcc9bb9b-swdvj",nf_instance_id="",operation_type="update",response_code="4xx",sbi_priority="",servicename_3gpp="npcf-smpolicycontrol",snssai="11-abc123",}

Table 9-32 ocpm_egress_request_total

Field	Details
Description	Total number of egress requests received.
Metric Type	Counter

Table 9-32 (Cont.) ocpm_egress_request_total

Field	Details
Dimensions	• app_kubernetes_io_instance • app_kubernetes_io_managed_by • app_kubernetes_io_name • app_kubernetes_io_part_of • app_kubernetes_io_version • application • container • endpoint • engVersion • exported_application • helm_sh_chart • hostname • instance • io_kompose_service • job • microservice • mktgVersion • namespace • nf_name • operation_type • pod • pod_template_hash • servicename_3gpp • vendor

Example:

ocpm_egress_request_total{app_kubernetes_io_instance = "ocpcf",app_kubernetes_io_managed_by = "Helm",app_kubernetes_io_name = "sm-service",app_kubernetes_io_part_of = "occnp",app_kubernetes_io_version = "1.0.0",application = "occnp",container = "sm-service",endpoint = "cnc-metrics",engVersion = "23.4.0",exported_application = "pcf_smservice",helm_sh_chart = "sm-service-23.4.0",hostname = "ocpcf-pcf-smservice-8576fc787-8z6zl",instance = "10.233.103.137:9000",io_kompose_service = "ocpcf-pcf-smservice",job = "occne-infra/occne-nf-cnc-podmonitor",microservice = "occnp_pcf_sm",mktgVersion = "1.0.0",namespace = "gg-pcf1",nf_name = "svc-perfgo.gg-perfgo-client",operation_type = "update_notify",pod = "ocpcf-pcf-smservice-8576fc787-8z6zl",pod_template_hash = "8576fc787",servicename_3gpp = "npcf-smpolicycontrol",vendor = "Oracle"} 1.0 ; Type-Counter

Table 9-33 ocpm_egress_response_total

Field	Details
Description	Total number of egress responses.
Metric Type	Counter

Table 9-33 (Cont.) ocpm_egress_response_total

Field	Details
Dimensions	• operation_type • dnn • snssai • nf_instance_id • sbi_priority • servicename_3gpp • response_code • latency

Examples:

- ocpm_egress_response_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="",nf_name="smf-oracle.com",latency="9",operation_type="update_notify",response_code="2xx",sbi_priority="",servicename_3gpp="npcf-smpolicycontrol",snssai="11-abc123",} 1.0 ; Type-Counter
- ocpm_egress_response_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="",nf_name="smf-oracle.com",latency="6",operation_type="terminate_notify",response_code="4xx",sbi_priority="",servicename_3gpp="npcf-smpolicycontrol",snssai="11-abc123",} 1.0 ; Type-Counter
- ocpm_egress_response_total{application="pcf_smservice",cause="PENDING_TRANSACTION",dnn="",hostname="jiandong-pcf-smservice-6dbcc9bb9b-swdvj",latency="0",nf_instance_id="",nf_name="",operation_type="update_notify",response_code="2xx",sbi_priority="",servicename_3gpp="npcf-smpolicycontrol",snssai="",} 1.0

Table 9-34 occnp_sm_binding_request_total

Field	Details
Description	Total number of binding requests.
Metric Type	Counter
Dimensions	• application • dnn • mode • operation_type • snssai

Examples:

- occnp_sm_binding_request_total{application="pcf_smservice",dnn="dnn1",hostname="a-pcf-smservice-fbbfccdf-vvdr6",mode="synchronous",nf_instance_id="fe7d992b-0541-4c7d-ab84-6d70b1babc1",nf_name="bsf.oracle.com",operation_type="create",snssai="11-abc111",} 1.0; Type-Counter
- occnp_sm_binding_request_total{application="pcf_smservice",dnn="dnn1",hostname="a-pcf-smservice-fbbfccdf-vvdr6",mode="synchronous",nf_instance_id="fe7d992b-0541-4c7d-ab84-6d70b1babc1",nf_name="bsf.oracle.com",operation_type="update",snssai="11-abc111",} 1.0; Type-Counter
- occnp_sm_binding_request_total{application="pcf_smservice",dnn="dnn1",hostname="a-pcf-smservice-fbbfccdf-vvdr6",mode="synchronous",nf_instance_id="fe7d992b-0541-4c7d-

ab84-6d70b1babc1",nf_name="bsf.oracle.com",operation_type="delete",snssai="11-abc111",} 1.0; Type-Counter

Table 9-35 occnp_sm_binding_response_total

Field	Details
Description	Total number of binding responses.
Metric Type	Counter
Dimensions	• application • dnn • mode • operation_type • snssai • response code

Examples:

- occnp_sm_binding_response_total{application="pcf_smservice",dnn="dnn1",hostname="a-pcf-smservice-fbbfccdf-vvdr6",mode="synchronous",nf_instance_id="fe7d992b-0541-4c7d-ab84-6d70b1babc1",nf_name="bsf.oracle.com",operation_type="create",response_code="2xx",snssai="11-abc111",} 1.0; Type-Counter
- occnp_sm_binding_response_total{application="pcf_smservice",dnn="dnn1",hostname="a-pcf-smservice-fbbfccdf-vvdr6",mode="synchronous",nf_instance_id="fe7d992b-0541-4c7d-ab84-6d70b1babc1",nf_name="bsf.oracle.com",operation_type="update",response_code="2xx",snssai="11-abc111",} 1.0; Type-Counter
- occnp_sm_binding_response_total{application="pcf_smservice",dnn="dnn1",hostname="a-pcf-smservice-fbbfccdf-vvdr6",mode="synchronous",nf_instance_id="fe7d992b-0541-4c7d-ab84-6d70b1babc1",nf_name="bsf.oracle.com",operation_type="delete",response_code="2xx",snssai="11-abc111",} 1.0; Type-Counter

Table 9-36 occnp_sm_binding_ex_total

Field	Details
Description	The number of binding operations that cannot be performed due to an exception (timeout).
Metric Type	Counter
Dimensions	• application • dnn • snssai • operation_type • mode

Examples:

- occnp_sm_binding_ex_total{application="pcf_smservice",dnn="dnn1",snssai="1-000001",operation_type="create",mode="asynchronous",} 1.0 ; Type-Counter
- occnp_sm_binding_ex_total{application="pcf_smservice",dnn="dnn1",snssai="1-000001",operation_type="update",mode="asynchronous",} 1.0 ; Type-Counter
- occnp_sm_binding_ex_total{application="pcf_smservice",dnn="dnn1",snssai="1-000001",operation_type="delete",mode="asynchronous",} 1.0 ; Type-Counter

Table 9-37 ocpm_egress_request_timeout_total

Field	Details
Description	Total number of request timeouts from SMF.
Metric Type	Counter
Dimensions	• application • dnn • egress_service • hostname • nf_instance_id • operation_type • resource_name • sbi_priority • service_name • snssai

Examples:

- ocpm_egress_request_timeout_total{application="pcf_smservice",dnn="dnn1",egress_service="SMF",hostname="pcf_smservice",nf_instance_id="",nf_name="smf-oracle.com",operation_type="POST",resource_name="SmPolicyNotification",sbi_priority="",service_name="notificaton",servicename_3gpp="npcf-smpolicycontrol",snssai="11-abc123",} 1.0; Type-Counter
- ocpm_egress_request_timeout_total{application="pcf_smservice",dnn="dnn1",egress_service="SMF",hostname="pcf_smservice",nf_instance_id="",nf_name="smf-oracle.com",operation_type="POST",resource_name="TerminationNotification",sbi_priority="",service_name="notificaton",servicename_3gpp="npcf-smpolicycontrol",snssai="11-abc123",} 1.0; Type-Counter
- ocpm_egress_request_timeout_total{application="pcf_smservice",dnn="dnn1",egress_service="BSF",hostname="pcf_smservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-6d70b1babc1",nf_name="bsf-oracle.com",operation_type="POST",resource_name="PCFSessionBindings",sbi_priority="",service_name="nbsf-management",servicename_3gpp="npcf-smpolicycontrol",snssai="11-abc123",} 1.0; Type-Counter
- ocpm_egress_request_timeout_total{application="pcf_smservice",dnn="dnn1",egress_service="BSF",hostname="pcf_smservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-6d70b1babc1",nf_name="bsf-oracle.com",operation_type="DELETE",resource_name="IndividualPCFSessionBinding",sbi_priority="",service_name="nbsf-management",servicename_3gpp="npcf-smpolicycontrol",snssai="11-abc123",} 1.0; Type-Counter

Table 9-38 remote_delete_notify_total

Field	Details
Description	Total number of notifications for session delete requests at SM service.
Metric Type	Counter

Table 9-38 (Cont.) remote_delete_notify_total

Field	Details
Dimensions	• Code • Request • SessionType • application • hostname

Examples:

- remote_delete_notify_total{Code="2xx",Request="DeleteNotify",SessionType="SM",application="pcf_smservice",hostname="4dfc2dfc25da",} 0.0
- remote_delete_notify_total{Code="4xx",Request="DeleteNotify",SessionType="SM",application="pcf_smservice",hostname="4dfc2dfc25da",} 0.0
- remote_delete_notify_total{Code="5xx",Request="DeleteNotify",SessionType="SM",application="pcf_smservice",hostname="4dfc2dfc25da",} 0.0
- remote_delete_notify_total{Code="N/A",Request="EmulateDelete",SessionType="SM",application="pcf_smservice",hostname="4dfc2dfc25da",} 0.0

Table 9-39 audit_notifications_sent

Field	Details
Description	Number of notifications sent by SM Service to SMF to check whether the session is stale or not.
Metric Type	Counter
Dimensions	Request

Table 9-40 audit_update_notify_session_not_found

Field	Details
Description	Number of 404 response sent by SMF for the records which are identified as stale by Audit Service.
Metric Type	Counter
Dimensions	Request

Table 9-41 audit_update_notify_session_found

Field	Details
Description	Number of 204 response sent by SMF for the records which are identified as stale by Audit Service.
Metric Type	Counter
Dimensions	Request

Table 9-42 audit_update_notify_response_error

Field	Details
Description	Number of update notify requests sent to SMF by SM service that were responded with an error code (except 404)
Metric Type	Counter
Dimensions	

Table 9-43 audit_update_notify_request_error

Field	Details
Description	Number of update notify requests that could not be sent to SMF or timed out
Metric Type	Counter
Dimensions	

Table 9-44 audit_update_timestamp_cnt

Field	Details
Description	Number of records whose LASTACESSTIME column is updated by SM Service when it receives 204 response from SMF
Metric Type	Counter
Dimensions	Request

Table 9-45 audit_delete_records_count

Field	Details
Description	Number of records deleted by SM Service when it receives 404 response from SMF
Metric Type	Counter
Dimensions	Request

Table 9-46 audit_delete_records_max_ttl_count

Field	Details
Description	Number of records deleted by SM Service when the max TTL was reached
Metric Type	Counter
Dimensions	

Table 9-47 session_oam_request_total

Field	Details
Description	Number of session requests for SM Service
Metric Type	Request

Table 9-47 (Cont.) session_oam_request_total

Field	Details
Dimensions	• Response Code • Operation Type • SessionType

Table 9-48 session_oam_response_total

Field	Details
Description	Number of session responses for SM Service
Metric Type	Response
Dimensions	• Response Code • Operation Type • SessionType

Table 9-49 occnp_http_sm_request_total

Field	Details
Description	Total number of http requests.
Metric Type	Request
Dimensions	• application • nf_instance_id • nfName • operationType • operation_type • service_name

Table 9-50 occnp_http_sm_response_total

Field	Details
Description	Total number of http responses.
Metric Type	Response
Dimensions	• application • nf_instance_id • nfName • operationType • operation_type • service_name

Table 9-51 occnp_http_in_conn_request_total

Field	Details
Description	
Metric Type	

Table 9-51 (Cont.) occnp_http_in_conn_request_total

Field	Details
Dimensions	• application • dnn • hostname • operationType • priority • sessRuleReports

Examples:

- occnp_http_in_conn_resquest_total{application="pcf_smservice",hostname="jiandong-pcf-smservice-5878554dc7-wgvc7",nfInstanceId="", ,operationType="create",priority="",servicename3gpp="npcf-smpolicycontrol",snssai="11-abc123",}

Table 9-52 occnp_http_in_conn_response_total

Field	Details
Description	
Metric Type	
Dimensions	• application • bindingId • bindingLevel • dnn • hostname • nfInstanceId • operationType • priority • responseCode • servicename3gpp • sessRuleReports • snssai • spendingLimitDataSource

Examples:

- occnp_http_in_conn_response_total{application="pcf_smservice",bindingId="",bindingLevel="",cause="PENDING_TRANSACTION",dnn="dnn1",hostname="jiandong-pcf-smservice-5878554dc7-wgvc7",nfInstanceId="", ,operationType="update",responseCode="400",priority="",servicename3gpp="npcf-smpolicycontrol",snssai="11-abc123",}

Table 9-53 occnp_http_out_conn_request_total

Field	Details
Description	Total number of http connection requests.
Metric Type	Counter

Table 9-53 (Cont.) occnp_http_out_conn_request_total

Field	Details
Dimensions	• application • bindingId • bindingLevel • hostname • nfInstanceId • operationType • priority • servicename3gpp • sessRuleReports • snssai

Examples:

- occnp_http_out_conn_request_total{application="pcf_smservice",bindingId="",bindingLevel="",dnn="dnn1",hostname="jiandong-pcf-smservice-5878554dc7-wgvc7",nfInstanceId="",nfName="",operationType="update_notify",priority="",retry="Yes",servicename3gpp="npcf-smpolicycontrol",snssai="",}

Table 9-54 occnp_http_out_conn_response_total

Field	Details
Description	Total number of http connection responses.
Metric Type	Counter
Dimensions	• application • cause • hostname • latency • nfInstanceId • operationType • responseCode • servicename3gpp • responseCode

Examples:

- occnp_http_out_conn_response_total{application="pcf_smservice",dnn="dnn1",cause="PENDING_TRANSACTION",hostname="jiandong-pcf-smservice-5878554dc7-wgvc7",latency="2",nfInstanceId="",nfName="",operationType="update_notify",responseCode="400",priority="",servicename3gpp="npcf-smpolicycontrol",snssai="",}

Table 9-55 occnp_sm_sess_rule_failure_total

Field	Details
Description	Indicates failure count because of session rule failure
Metric Type	Counter

Table 9-55 (Cont.) occnp_sm_sess_rule_failure_total

Field	Details
Dimensions	• application • failure • hostname • operationType • rule_id • rule_status

Table 9-56 lock_request_total

Field	Details
Description	Number of times the SM Service tries to acquire the lock
Metric Type	Counter
Dimensions	• application • podName • requestType • serviceName

Table 9-57 lock_response_total

Field	Details
Description	Number of times the Bulwark Service responds to SM Service for the lock request
Metric Type	Counter
Dimensions	• application • podName • requestType • serviceName • responseStatus • responseType

Table 9-58 occnp_http_sm_request_total

Field	Details
Description	Number of times the SM Service sends an update request
Metric Type	Counter
Dimensions	• application • nfInstanceId • nfName • operationType • request • servicenameNon3gpp

Table 9-59 occnp_http_sm_response_total

Field	Details
Description	Number of times the SM Service receives a response
Metric Type	Counter
Dimensions	• application • latency • nfInstanceId • nfName • operationType • request • responseCode • servicenameNon3gpp

Table 9-60 occnp_lease_duration_seconds

Field	Details
Description	Time in seconds for which SM Service holds the lock
Metric Type	Histogram
Dimensions	• Lock_Lease_Duration • Message_Type • application • quantile

Table 9-61 rule_action

Field	Details
Description	Provides metric for: • `SUCC_RES_ALLO` sent in `lastreqruledata` • `SUCC_RES_ALLO` received from SMF
Metric Type	
Dimensions	• ruleType • Type • Action

Examples:

-
- rule_action_total{action="SUCC_RES_ALLO",application="pcf_smservice",rule_type="pcc",type="request",} 1.0
- rule_action_total{action="SUCC_RES_ALLO",application="pcf_smservice",rule_type="pcc",type="report",} 1.0

Table 9-62 ocpm_timeout_total

Field	Details
Description	Total number of increments whenever a timeout is detected
Metric Type	Counter
Dimensions	• operation • interface

Examples:

- ocpm_timeout_total{application="pcf_amservice",hostname="pdmayanspolicy-pcf-amservice-6b64cbdd8b-pzxkq",operationType="query_subs",servicename3gpp="npcf-am-policy-control",targetService="USER_SERVICE",} 1.0

Table 9-63 topic_version

Field	Details
Description	Describes the current applied version of a given topic (mentioned in dimension topic_name) into the pod.
Metric Type	Gauge
Dimensions	• Service Name • Pod Name

Table 9-64 occnp_db_overall_processing_time

Field	Details
Description	Overall processing time of a database.
Metric Type	Counter
Dimensions	• Table • Method • status • compressionScheme

Examples:

- occnp_db_overall_processing_time_seconds_max{Method="put",Status="Success",Table="SmPolicyAssociation",application="pcf_smservice",compressionScheme="ZLIB_COMPRESSION_APPLICATION",} 0.314998

Table 9-65 audit_delete_records_max_ttl_count

Field	Details
Description	Indicates number of audit notifications sent by Audit service to SM service that are processed with MaxTTL as true (delete Rx Session, NO ASR).
Metric Type	Counter

Table 9-65 (Cont.) audit_delete_records_max_ttl_count

Field	Details
Dimensions	• SvcType (SM Service, PA Service) • requestType

Table 9-66 audit_notifications_sent

Field	Details
Description	Number of audit notifications to SM service that are processed.
Metric Type	Counter
Dimensions	SvcType (PA Service)

Table 9-67 audit_delete_records_count

Field	Details
Description	Indicates the number of deletions after failure ASA.
Metric Type	Counter
Dimensions	• SvcType (PA Service) • request (TerminateNotify)

Table 9-68 audit_terminate_notify

Field	Details
Description	ASA response code metrics
Metric Type	Counter
Dimensions	• SvcType (PA Service) • request (TerminateNotify) • responseCode (2xx, 4xx, or 5xx)

Table 9-69 occnp_internal_delete_request_total

Field	Details
Description	Total number of intenal delete request triggered on remote cleanup when forceDeleteOnError is true or forceDeleteOnExpiryOfWaitTimer is true.
Metric Type	Counter
Dimensions	• application • cause (waitforSMDeleteExpired or forceDeleteOnErrortoNotifySMF

Examples:

- occnp_internal_delete_request_total{application="pcf_smservice",cause="waitforSMDeleteExpired",} 1.0
- occnp_internal_delete_request_total{application="pcf_smservice",cause="forceDeleteOnErrortoNotifySMF",} 1.0

Table 9-70 occnp_internal_delete_response_total

Field	Details
Description	Total number of internal delete response when forceDeleteOnError is true or forceDeleteOnExpiryOfWaitTimer is true.
Metric Type	Counter
Dimensions	• application • response_code

Examples:

- occnp_internal_delete_response_total{application="pcf_smservice",response_code="204 NO_CONTENT",} 1.0
- occnp_internal_delete_response_total{application="pcf_smservice",response_code="404 NOT_FOUND",} 1.0

Table 9-71 error_handler_in_total

Field	Details
Description	This metric is incremented on initialization of error handling.
Type	Counter
Dimension	• application • application_exception • hostname • wrapped_exception

Examples:

- error_handler_in_total{application="pcf_smservice",application_exception="DiameterException",hostname="rigelns2-pcf-smservice-d846c5cdb-ps5s9",wrapped_exception="DiameterClientResponseException",} 1.0

Table 9-72 error_handler_exec_total

Field	Details
Description	This metric is incremented on error handling performed by framework.
Type	Counter
Dimension	• application • application_exception • error_type • operation • origin • rule_name • source_interface • target_interface • wrapped_exception

Examples:

- error_handler_exec_total{application="pcf_smservice",application_exception="DiameterException",error_type="INTERNAL",hostname="gi-frank-pcf-pcf-smservice-6496bc6d6d-

n4v4p",operation="RAR",origin="DIAMETER",rule_name="TERMINATE_TRANSACTION",source_interface="RX",status="5010",target_interface="ANY",wrapped_exception="DiameterClientResponseException",} 1.0

Table 9-73 error_handler_out_total

Field	Details
Description	This metric is incremented on completion of error handling.
Type	Counter
Dimension	• application • application_exception • error_resolved • hostname • wrapped_exception

Examples:

- error_handler_out_total{application="pcf_smservice",application_exception="DiameterException",error_resolved="false",hostname="rigelns2-pcf-smservice-d846c5cdb-ps5s9",wrapped_exception="DiameterClientResponseException",} 1.0

Table 9-74 occnp_pcf_smservice_overall_processing_time_seconds

Field	Details
Description	SM service overall processing time.
Type	Summary
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Table 9-75 occnp_pcf_smservice_overall_processing_time_seconds_max

Field	Details
Description	SM service overall processing time.
Type	Gauge
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Table 9-76 occnp_pcf_smservice_overall_processing_time_seconds_count

Field	Details
Description	SM service overall processing time.
Type	Counter
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Table 9-77 update_notify_total

Field	Details
Description	This metrics is pegged for to show that there is no notification being sent when there is an AAR message with the flow status as REMOVED and the Message type as either AUDIO, MESSAGE, or OTHER.
Type	Counter
Dimension	• application • cause • hostname • optimized
Example	update_notify_total{application="pcf_smservice",cause="FLOW_STATUS_REMOVED",hostname="gmtommy-pcf-smservice-56cb4584df-clsfd",optimized="Yes",} 2.0

Table 9-78 collision_detection_local_delete

Field	Details
Description	Measures number of internal deletes due to duplicate session request.
Type	Counter
Dimension	• ServiceType • Request
Example	

Table 9-79 collision_detection_terminate_notify

Field	Details
Description	Measures number of session terminate request sent due to duplicate session deletion.
Type	Counter
Dimension	• ServiceType • Request
Example	

Table 9-80 audit_max_ttl_terminate_notify

Field	Details
Description	Measures number of session terminate request sent due to Stale Session Deletion Request
Type	Counter
Dimension	• ServiceType • Request
Example	

Table 9-81 occnp_feature_info_received_total

Field	Details
Description	Measures sessRuleReports and policyDecFailureReports in request and response.
Type	Counter
Dimension	• application • hostname • servicename3gpp • operationType • responseCode • sessRuleReports • policyDecFailureReports
Example	occnp_feature_info_received_total{application="pcf_smservice",hostname="sky8-pcf-smservice-75df58b555-srm7q",operationType="update",policyDecFailureReports="true",responseCode="",servicename3gpp="npcf-smpolicycontrol",sessRuleReports="false",} 2.0 occnp_feature_info_received_total{application="pcf_smservice",hostname="sky8-pcf-smservice-75df58b555-srm7q",operationType="update_notify",policyDecFailureReports="true",responseCode="200",servicename3gpp="npcf-smpolicycontrol",sessRuleReports="false",} 2.0

Table 9-82 error_handler_in_total

Field	Details
Description	This metric is incremented on initialization of error handling.
Type	Counter
Dimensions	• application • applicationException • wrapped_exception
Example	occnp_error_handler_in_total{application="pcf_smservice",application_exception="HttpException",status="404",wrapped_exception="NotFound",} 18.0

Table 9-83 error_handler_exec_total

Field	Details
Description	This metric is incremented on error handling performed by framework.
Type	Counter

Table 9-83 (Cont.) error_handler_exec_total

Field	Details
Dimensions	• application • applicationException • error_type • operation • origin • rule_name • source_interface • target_interface • wrapped_exception
Example	occnp_error_handler_exec_total{application="pcf_smservice",application_exception="JavaException",error_type="INTERNAL",operation="UPDATE",origin="JAVA",rule_name="REJECT_WITH_ENHANCED_DETAIL",source_interface="SMF",status="404",target_interface="POLICY",wrapped_exception="ServiceException",} 1.0

Table 9-84 error_handler_out_total

Field	Details
Description	This metric is incremented on completion of error handling.
Type	Counter
Dimensions	• application • applicationException • error_resolved • wrapped_exception
Example	occnp_error_handler_out_total{application="pcf_smservice",application_exception="HttpException",error_resolved="false",status="423",wrapped_exception="WebClientResponseException",} 4.0

For more information about Dimensions, see CNC Policy Metrics.

9.7 AM Service Metrics

The following table describes the AM Service metrics and respective dimensions:

Table 9-85 ocpm_ingress_request_total

Field	Details
Description	Total number of ingress requests received.
Metric Type	Counter
Dimensions	• operation_type • nf_instance_id • sbi_priority • service name_ 3gpp

Table 9-85 (Cont.) ocpm_ingress_request_total

Field	Details
Example	: 1. ocpm_ingress_request_total{operation_type="create",servicename_3gpp="npcf-am-policy-control/v1",} 2.0 ; Type-Counter 2. ocpm_ingress_request_total{operation_type="get",servicename_3gpp="npcf-am-policy-control/v1",} 1.0 ; Type-Counter

Table 9-86 ocpm_ingress_response_total

Field	Details
Description	Total number of ingress responses.
Metric Type	Counter
Dimensions	• Cause • operation_type • nf_instance_id • sbi_priority • servicename_3gpp • response_code
Example	• ocpm_ingress_response_total{operation_type="create",response_code="2xx",servicename_3gpp="npcf-am-policy-control/v1",} 2.0 ; Type-Counter • ocpm_ingress_response_total{operation_type="get",response_code="2xx",servicename_3gpp="npcf-am-policy-control/v1",} 1.0 ; Type-Counter

Table 9-87 ocpm_egress_request_total

Field	Details
Description	Total number of egress requests received.
Metric Type	Counter
Dimensions	• operation_type • nf_instance_id • sbi_priority • servicename_3gpp
Example	• ocpm_egress_request_total{operation_type="update_notify",servicename_3gpp="npcf-am-policy-control/v1",} 2.0 ; Type-Counter • ocpm_egress_request_total{operation_type="terminate_notify",servicename_3gpp="npcf-am-policy-control/v1",} 1.0 ; Type-Counter

Table 9-88 ocpm_egress_response_total

Field	Details
Description	Total number of egress responses.
Metric Type	Counter
Dimensions	• operation_type • nf_instance_id • sbi_priority • servicename_3gpp • response_code • latency
Example	• ocpm_egress_response_total{operation_type="terminate_notify",response_code="2xx",servicename_3gpp="npcf-am-policy-control/v1",} 1.0 ; Type-Counter • ocpm_egress_response_total{operation_type="update_notify",response_code="2xx",servicename_3gpp="npcf-am-policy-control/v1",} 2.0 ; Type-Counter

Table 9-89 am_audit_update_notify_sent_total

Field	Details
Description	Total number of audit notifications sent.
Metric Type	Counter
Dimensions	Request
Example	• am_audit_update_notify_sent_total{} 1.0 ; Type-Counter

Table 9-90 am_audit_update_notify_session_not_found_total

Field	Details
Description	Total number of 'session not found' notifications.
Metric Type	Counter
Dimensions	Request
Example	• am_audit_update_notify_session_not_found_total{} 1.0 ; Type-Counter

Table 9-91 am_audit_update_notify_session_found_total

Field	Details
Description	Total number of 'session found' notifications.
Metric Type	Counter
Dimensions	Request
Example	• am_audit_update_notify_session_found_total{} 1.0 ; Type-Counter

Table 9-92 am_audit_update_timestamp_cnt_total

Field	Details
Description	Total number of timestamp countfor audit requests.
Metric Type	Counter
Dimensions	Request
Example	• am_audit_update_timestamp_cnt_total{} 1.0 ; Type-Counter

Table 9-93 am_audit_delete_records_count_total

Field	Details
Description	Total number of delete audit records requests
Metric Type	Counter
Dimensions	Request
Example	• am_audit_delete_records_count_total{notification="true",} 1.0 ; Type-Counter • am_audit_delete_records_count_total{notification="false",} 1.0 ; Type-Counter

Table 9-94 am_audit_delete_records_max_ttl_count_total

Field	Details
Description	Number of records deleted by AM service when the max TTL was reached.
Metric Type	Counter
Dimensions	Request
Example	• am_audit_delete_records_max_ttl_count_total{} 1.0 ; Type-Counter

Table 9-95 am_audit_amf_query_total

Field	Details
Description	
Metric Type	Counter
Dimensions	Request
Example	• am_audit_amf_query_total{} 1.0 ; Type-Counter

Table 9-96 am_audit_update_notify_response_error_total

Field	Details
Description	Number of update notify requests sent to AMF by AM service that were responded with an error code.
Metric Type	Counter

Table 9-96 (Cont.) am_audit_update_notify_response_error_total

Field	Details
Dimensions	Request
Example	• am_audit_update_notify_response_error_total{} 1.0 ; Type-Counter

Table 9-97 session_oam_request_total

Field	Details
Description	Number of session requests for AM Service
Metric Type	Request
Dimensions	• Response Code • Operation Type • Session Type

Table 9-98 session_oam_response_total

Field	Details
Description	Number of session responses for AM Service
Metric Type	Response
Dimensions	• Response Code • Operation Type • Session Type

Table 9-99 session_info_request_total

Field	Details
Description	Total number of sessions to be deleted as requested by the AM Service
Metric Type	Counter
Dimensions	• Application • Cause • Operation Type • Session Type
Example	• session_info_request_total{application="pcf_amservice",cause="local_limiting_session",operation_type="delete",session_type="am_session",} 3.0

Table 9-100 session_info_response_total

Field	Details
Description	Total number of sessions deleted as requested by the AM Service
Metric Type	Counter

Table 9-100 (Cont.) session_info_response_total

Field	Details
Dimensions	• Application • Cause • Operation Type • Response Code • Session Type
Example	• session_info_response_total{application="pcf_amservice",cause="local_limiting_session",operation_type="delete",response_code="5xx",session_type="am_session",} 2.0

Table 9-101 http_out_conn_request_total

Field	Details
Description	Total number of connection requests recieved
Metric Type	Counter
Dimensions	• Application • bindingId • bindingLevel • discoveryId • discoveryParameter • hostname • operationType
Example	• http_out_conn_request_total{application="pcf_amservice",bindingId="",bindingLevel="",discoveryId="set001.region01.amfset.5gc.mnc012.mcc345",discoveryParameter="target-nf-set-id",hostname="my-cnpolicy-pcf-amservice-6c54cd7d4d-kp295",operationType="get",servicename3gpp="npcf-am-policy-control",} 2.0

Table 9-102 http_out_conn_response_total

Field	Details
Description	Total number of connection response sent
Metric Type	Counter
Dimensions	• Application • hostname • operationType • responseCode
Example	• http_out_conn_response_total{application="pcf_amservice",hostname="my-cnpolicy-pcf-amservice-6c54cd7d4d-kp295",operationType="update_notify",responseCode="5xx",servicename3gpp="npcf-am-policy-control",} 2.0

Table 9-103 ocpm_ar_request_total

Field	Details
Description	Total number of ar requests recieved
Metric Type	Counter
Dimensions	• Application • fqdn • hostname • scheme • servicename3gpp
Example	• ocpm_ar_request_total{application="pcf_amservice",fqdn="nf12stub.harald-ns.svc",hostname="my-cnpolicy-pcf-amservice-6c54cd7d4d-kp295",scheme="http",servicename3gpp="npcf-am-policy-control",} 1.0 • ocpm_ar_request_total{application="pcf_amservice",fqdn="nf12stub.harald-ns.svc",hostname="my-cnpolicy-pcf-amservice-6c54cd7d4d-kp295",scheme="http",servicename3gpp="npcf-am-policy-control",} 1.0

Table 9-104 ocpm_ar_response_total

Field	Details
Description	Total number of ar response sent
Metric Type	Counter
Dimensions	• Application • fqdn • hostname • scheme • servicename3gpp

Table 9-105 occnp_http_in_conn_response_total

Field	Details
Description	Number of http responses for AM Service
Metric Type	Counter
Dimensions	• application • bindingId • bindingLevel • hostname • nfInstanceId • operationType • responseCode • priority • servicename3gpp

Table 9-105 (Cont.) occnp_http_in_conn_response_total

Field	Details
Example	• **occnp_http_in_conn_response_total**{application="pcf_amservice",bindingId="",bindingLevel="",dnn="dnn1",hostname="jiandong-pcf-smservice-5878554dc7-wgvc7",nfInstanceId="", ,operationType="update",responseCode="400",priority="",servicename3gpp="npcf-amservice",snssai="11-abc123",***cause="PENDING_TRANSACTION"***

Table 9-106 http_bulwark_lock_request_total

Field	Details
Description	Number of bulwark lock requests
Metric Type	Counter
Dimensions	• application • hostname • operationType • servicenameNon3gpp
Example	• **http_bulwark_lock_request_total**{application="pcf_amservice",hostname="jiandong-pcf-smservice-5878554dc7-wgvc7",operationType="**bulwark_lock**",servicenameNon3gpp="npcf-amservice"}

Table 9-107 http_bulwark_unlock_request_total

Field	Details
Description	Number of bulwark unlock requests
Metric Type	Counter
Dimensions	• application • hostname • operationType • servicenameNon3gpp
Example	• **http_bulwark_unlock_request_total**{application="pcf_amservice",hostname="jiandong-pcf-smservice-5878554dc7-wgvc7",operationType="**bulwark_unlock**",servicenameNon3gpp="npcf-amservice"}

Table 9-108 http_bulwark_lock_response_total

Field	Details
Description	Number of bulwark lock responses
Metric Type	Counter

Table 9-108 (Cont.) http_bulwark_lock_response_total

Field	Details
Dimensions	• application • hostname • operationType • servicenameNon3gpp
Example	• **http_bulwark_lock_response_total**{application="pcf_amservice",hostname="jiandong-pcf-smservice-5878554dc7-wgvc7",operationType="**bulwark_lock**",responseCode="201",servicenameNon3gpp="npcf-smpolicycontrol"}

Table 9-109 http_bulwark_lock_request_retry_total

Field	Details
Description	Number of bulwark lock request retries
Metric Type	Counter
Dimensions	• application • hostname • operationType • servicenameNon3gpp
Example	• **http_bulwark_lock_request_retry_total**{application="pcf_amservice",hostname="jiandong-pcf-smservice-5878554dc7-wgvc7",operationType="**bulwark_lock**",responseCode="204",servicenameNon3gpp="npcf-amservice"}

Table 9-110 http_bulwark_unlock_request_retry_total

Field	Details
Description	Number of bulwark unlock request retries
Metric Type	Counter
Dimensions	• application • hostname • operationType • servicenameNon3gpp
Example	• **http_bulwark_unlock_request_retry_total**{application="pcf_amservice",hostname="jiandong-pcf-smservice-5878554dc7-wgvc7",operationType="**bulwark_unlock**",responseCode="204",servicenameNon3gpp="npcf-amservice"}

Table 9-111 occnp_http_out_conn_request_total

Field	Details
Description	Number of http requestd for AM Service

Table 9-111 (Cont.) occnp_http_out_conn_request_total

Field	Details
Metric Type	Counter
Dimensions	• application • bindingId • bindingLevel • hostname • nfInstanceId • operationType • responseCode • priority • servicename3gpp
Example	• **occnp_http_out_conn_request_total**{application="pcf_amservice",bindingId="",bindingLevel="",dnn="dnn1",hostname="jiandong-pcf-smservice-5878554dc7-wgvc7",nfInstanceId="",nfName="",operationType="update_notify",priority="",servicename3gpp="npcf-amservice",snssai="",***retry="Yes"***}

Table 9-112 topic_version

Field	Details
Description	Describes the current applied version of a given topic (mentioned in dimension topic_name) into the pod.
Metric Type	Gauge
Dimensions	• Service Name • Pod Name

Table 9-113 occnp_db_overall_processing_time_seconds_count

Field	Details
Description	This metric is incremented when PCF performs GET, INSERT, UPDATE, and DELETE operation on an Association
Metric Type	Counter
Dimensions	• Method • Status • Table
Example	• occnp_db_overall_processing_time_seconds_count{Method="delete", Status="Success", Table="UePolicyAssociation", container="ue-service"}

Table 9-114 occnp_db_overall_processing_time_seconds_sum

Field	Details
Description	This metric is incremented when PCF performs DELETE operation on an Association
Metric Type	Counter
Dimensions	• Method • Status • Table
Example	• occnp_db_overall_processing_time_seconds_sum{Method="delete", Status="Success", Table="UePolicyAssociation", container="ue-service"}

Table 9-115 occnp_db_overall_processing_time_seconds_sum

Field	Details
Description	This metric is incremented when PCF performs DELETE operation on an Association
Metric Type	Counter
Dimensions	• Method • Status • Table
Example	• occnp_db_overall_processing_time_seconds_sum{Method="delete", Status="Success", Table="UePolicyAssociation", container="ue-service"}

Table 9-116 occnp_amservice_overall_processing_time_seconds_max

Field	Details
Description	This metric is incremented for AM service to calculate processing's duration of HTTP request arrived on rest controller. **Note**: occnp_amservice_overall_processing_time_seconds_max metric replaces http_server_requests_seconds_max.
Metric Type	Counter
Dimensions	• Method • Status • uri • container
Example	• occnp_amservice_overall_processing_time_seconds_max{method="DELETE", status="204", uri="/npcf-am-policy-control/v1/policies/{polAssoId}", container="am-service"}

Table 9-117 occnp_amservice_overall_processing_time_seconds_sum

Field	Details
Description	This metric is incremented for AM service to calculate processing's duration of HTTP request arrived on rest controller. **Note**: occnp_amservice_overall_processing_time_seconds_sum metric replaces http_server_requests_seconds_sum.
Metric Type	Counter
Dimensions	• Method • Status • uri • container
Example	• occnp_amservice_overall_processing_time_seconds_max{method="DELETE", status="204", uri="/npcf-am-policy-control/v1/policies/{polAssoId}", container="am-service"}

Table 9-118 occnp_amervice_overall_processing_time_seconds_count

Field	Details
Description	This metric is incremented for AM service to calculate processing's duration of HTTP request arrived on rest controller. **Note**: occnp_amservice_overall_processing_time_seconds_count metric replaces http_server_requests_seconds_count.
Metric Type	Counter
Dimensions	• Method • Status • uri • container
Example	• occnp_amservice_overall_processing_time_seconds_max{method="DELETE", status="204", uri="/npcf-am-policy-control/v1/policies/{polAssoId}", container="am-service"}

Table 9-119 error_handler_in_total

Field	Details
Description	This metric is incremented on initialization of error handling.
Type	Counter
Dimensions	• application • applicationException • wrapped_exception
Example	occnp_error_handler_in_total{application="pcf_amservice",application_exception="HttpException",status="404",wrapped_exception="NotFound",} 1.0

Table 9-120 error_handler_exec_total

Field	Details
Description	This metric is incremented on error handling performed by framework.
Type	Counter
Dimensions	• application • applicationException • error_type • operation • origin • rule_name • source_interface • target_interface • wrapped_exception
Example	occnp_error_handler_exec_total{application="pcf_amservice",application_exception="JavaException",error_type="INTERNAL",operation="CREATE",origin="JAVA",rule_name="REJECT_WITH_ENHANCED_DETAIL",source_interface="AMF",status="400",target_interface="POLICY",wrapped_exception="ServiceException",} 2.0

Table 9-121 error_handler_out_total

Field	Details
Description	This metric is incremented on completion of error handling.
Type	Counter
Dimensions	• application • applicationException • error_resolved • wrapped_exception
Example	occnp_error_handler_out_total{application="pcf_amservice",application_exception="HttpException",error_resolved="false",status="404",wrapped_exception="NotFound",} 1.0

Note

For more information about Dimensions, see CNC Policy Metrics.

9.8 CM Service Metrics

The following table describes the CM Service metrics and respective dimensions:

Table 9-122 system_operational_state

Field	Details
Description	This metric indicates the current operational state
Metric Type	Gauge

Table 9-122 (Cont.) system_operational_state

Field	Details
Dimensions	• application • eng_version • namespace • node • pod • vendor

For more information about Dimensions, see CNC Policy Metrics.

Examples

- system_operational_state{application="config-server",eng_version="",microservice="",namespace="",node="",pod="occnp-config-mgmt",vendor="oracle",} 1.0

9.9 PA Service Metrics

The following table describes the PA Service metrics and respective dimensions:

Table 9-123 ocpm_ingress_request_total

Field	Details
Description	Total number of ingress requests received.
Metric Type	Counter
Dimensions	• application • dnn • nf_instance_id • operation_type • sbi_priority • servicename_3gpp • snssai

Examples:

-
- ocpm_ingress_request_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="",operation_type="create",sbi_priority="",servicename_3gpp="npcf-policyauthorization",snssai="11-abc123",} 1.0 ; Type-Counter
- ocpm_ingress_request_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="",operation_type="update",sbi_priority="",servicename_3gpp="npcf-policyauthorization",snssai="11-abc123",} 1.0 ; Type-Counter
- ocpm_ingress_request_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="",operation_type="delete",sbi_priority="",servicename_3gpp="npcf-policyauthorization",snssai="11-abc123",} 1.0 ; Type-Counter
- ocpm_ingress_request_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="",operation_type="get",sbi_priority="",servicename_3gpp="npcf-policyauthorization",snssai="11-abc123",} 1.0 ; Type-Counter

Table 9-124 ocpm_ingress_response_total

Field	Details
Description	Total number of ingress responses.
Metric Type	Counter
Dimensions	• application • dnn • nf_instance_id • operation_type • response_code • sbi_priority • servicename_3gpp • snssai • binding_level • binding_id

Examples:

- ocpm_ingress_response_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="", operation_type="create",response_code="2xx",sbi_priority="",servicename_3gpp="npcf-policyauthorization",snssai="11-abc123",} 1.0 ; Type-Counter
- ocpm_ingress_response_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="", operation_type="update",response_code="2xx",sbi_priority="",servicename_3gpp="npcf-policyauthorization",snssai="11-abc123",binding_level="", binding_id=""} 1.0 ; Type-Counter
- ocpm_ingress_response_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="", operation_type="delete",response_code="4xx",sbi_priority="",servicename_3gpp="npcf-policyauthorization",snssai="11-abc123",binding_level="", binding_id=""} 1.0 ; Type-Counter
- ocpm_ingress_response_total{application="pcf_smservice",dnn="dnn1",nf_instance_id="", operation_type="get",response_code="2xx",sbi_priority="",servicename_3gpp="npcf-policyauthorization",snssai="11-abc123",binding_level="", binding_id=""} 1.0 ; Type-Counter

Table 9-125 remote_delete_notify_total

Field	Details
Description	Total number of delete requests.
Metric Type	Counter
Dimensions	• Code • Request • SessionType • hostname

Examples:

- remote_delete_notify_total{Code="2xx",Request="DeleteNotify",SessionType="PA",application="pcf_smservice",hostname="4dfc2dfc25da",} 0.0
- remote_delete_notify_total{Code="4xx",Request="DeleteNotify",SessionType="PA",application="pcf_smservice",hostname="4dfc2dfc25da",} 0.0

- remote_delete_notify_total{Code="5xx",Request="DeleteNotify",SessionType="PA",application="pcf_smservice",hostname="4dfc2dfc25da",} 0.0
- remote_delete_notify_total{Code="N/A",Request="EmulateDelete",SessionType="PA",application="pcf_smservice",hostname="4dfc2dfc25da",} 0.0

Table 9-126 session_oam_request_total

Field	Details
Description	Number of session requests for PA Service
Metric Type	Request
Dimensions	• Response Code • Operation Type • SessionType

Table 9-127 session_oam_response_total

Field	Details
Description	Number of session responses for PA Service
Metric Type	Response
Dimensions	• Response Code • Operation Type • SessionType

For more information about Dimensions, see CNC Policy Metrics.

9.10 UE Service Metrics

The following table describes the new UE Service metrics and the respective dimensions:

Table 9-128 ue_n1_ursp_rule_delivery

Field	Details
Description	This metric is incremented when a URSP rule is selected for delivery to UE.
Metric Type	Counter
Dimensions	ruleName

Table 9-129 ue_n1_transfer_request_total

Field	Details
Description	This metric shall be incremented when PCF sends an encoded N1 message (fragment) to AMF for delivery to UE.
Metric Type	Counter
Dimensions	• fragCount • fragIdx

Table 9-130 ue_n1_transfer_response_total

Field	Details
Description	This metric shall be incremented when PCF receives a response from AMF for an N1 message (fragment) delivery to UE.
Metric Type	Counter
Dimensions	• fragCount • fragIdx • responseCode • latency • cause

Table 9-131 ue_n1_transfer_exception

Field	Details
Description	This metric shall be incremented when PCF encounters an exception and fails to initiate an N1 message (fragment) delivery.
Metric Type	Counter
Dimensions	• fragCount • fragIdx

Table 9-132 ue_n1_transfer_retransmit

Field	Details
Description	This metric shall be incremented when PCF re-transmits an encoded N1 message (fragment) to AMF for delivery to UE.
Metric Type	Counter
Dimensions	• fragCount • fragIdx

Table 9-133 ue_n1_transfer_abort

Field	Details
Description	This metric shall be incremented when PCF aborts an N1 message delivery procedure.
Metric Type	Counter
Dimensions	reason

Table 9-134 ue_n1_transfer_skip

Field	Details
Description	This metric shall be incremented when PCF skips an N1 message fragment.
Metric Type	Counter
Dimensions	NA

Table 9-135 ue_n1_transfer_failure_notification

Field	Details
Description	This metric shall be incremented when PCF receives a transfer failure notification from the AMF.
Metric Type	Counter
Dimensions	cause

Table 9-136 ue_n1_transfer_ue_notification

Field	Details
Description	This metric shall be incremented when PCF receives an N1 notification from the UE.
Metric Type	Counter
Dimensions	commandType

The following table describes the UE Service metrics and respective dimensions:

Table 9-137 ocpm_ingress_request_total

Field	Details
Description	Total number of ingress requests received
Metric Type	Counter
Dimensions	• operation_type • service name_ 3gpp
Example	• ocpm_ingress_request_total{operation_type="get",servicename_3gpp="npcf-ue-policy-control",} 2.0 ; Type-Counter

Table 9-138 ocpm_ingress_response_total

Field	Details
Description	Total number of ingress responses
Metric Type	Counter
Dimensions	• cause • operation_type • servicename_3gpp • response_code
Example	• ocpm_ingress_response_total{operation_type="get",response_code="5xx",servicename_3gpp="npcf-ue-policy-control",} 4.0 ; Type-Counter • ocpm_ingress_response_total{cause="Internal Server Error",operation_type="get",response_code="5xx",servicename_3gpp="npcf-ue-policy-control",} 4.0 ; Type-Counter

Table 9-139 ocpm_egress_request_total

Field	Details
Description	Total number of egress requests received
Metric Type	Counter
Dimensions	• operation_type • dnn • snssai • nf_instance_id • sbi_priority • servicename_3gpp
Example	• ocpm_egress_request_total{application="pcf_ueservice",binding_id="",binding_level="",discovery_id="",discovery_parameter="target-nf-set-id",hostname="pcf-pcf-ueservice-7988ccfbd9-zn8hs",nf_instance_id="13515195-c537-4645-9b97-96ec797fbbbf",nf_name="nf1stub.gunther-ns.svc",operation_type="audit_notification",servicename_3gpp="npcf-ue-policy-control",} 1.0

Table 9-140 ocpm_egress_response_total

Field	Details
Description	Total number of egress responses.
Metric Type	Counter
Dimensions	• operation_type • dnn • snssai • nf_instance_id • sbi_priority • servicename_3gpp • response_code • latency
Example	• ocpm_egress_response_total{application="pcf_ueservice",hostname="pcf-pcf-ueservice-7988ccfbd9-zn8hs",nf_instance_id="13515195-c537-4645-9b97-96ec797fbbbf",nf_name="nf1stub.gunther-ns.svc",operation_type="audit_notification",response_code="5xx",servicename_3gpp="npcf-ue-policy-control",} 1.0

Table 9-141 session_viewer_request_total

Field	Details
Description	This metric shows the total number of query requests made using the Session Viewer.
Metric Type	Counter

Table 9-141 (Cont.) session_viewer_request_total

Field	Details
Dimensions	• operation_type (GET or DELETE) • response_code (2xx, 5xx) • session_type(ue_session)

Table 9-142 session_viewer_response_total

Field	Details
Description	This metric shows the total number of responses to query requests made using the Session Viewer.
Metric Type	Counter
Dimensions	

Table 9-143 session_oam_request_total

Field	Details
Description	Number of session requests for UE Service
Metric Type	Request
Dimensions	• Response Code • Operation Type • SessionType

Table 9-144 session_oam_response_total

Field	Details
Description	Number of session responses for UE Service
Metric Type	Response
Dimensions	• Response Code • Operation Type • SessionType

Table 9-145 occnp_ue_audit_stale_records_count_total

Field	Details
Description	Total number of stale records received from Audit Service in all notifications
Metric Type	Counter
Dimensions	• application • hostname
Example	• occnp_ue_audit_stale_records_count_total{application="pcf_ueservice",hostname="pcf-pcf-ueservice-58f8bd57d5-frggs",} 1.0

Table 9-146 occnp_ue_audit_deleted_records_count_total

Field	Details
Description	Total number of stale records deleted.
Metric Type	Counter
Dimensions	• application • hostname
Example	• occnp_ue_audit_deleted_records_count_total{application="pcf_ueservice",hostname="pcf-pcf-ueservice-58f8bd57d5-frggs",} 1.0

Table 9-147 occnp_ue_update_notify_rcvd_error_response_total

Field	Details
Description	Total number of update notify requests that can not be sent to AMF or timed out
Metric Type	Counter
Dimensions	• application • hostname • response_code
Example	• occnp_ue_update_notify_rcvd_error_response_total{application="pcf_ueservice",hostname="pcf-pcf-ueservice-7988ccfbd9-zn8hs",response_code="5xx",} 1.0

Table 9-148 audit_recs_stale_total

Field	Details
Description	Total number of records detected as stale
Metric Type	Counter
Dimensions	• ServiceName • TableName
Example	• audit_recs_stale_total{ServiceName="pcf-ueservice",TableName="UePolicyAssociation",} 1.0

Table 9-149 occnp_ue_audit_notif_sent_to_amf_count_total

Field	Details
Description	Total number of notifications sent to AMF on receiving stale session notification from audit
Metric Type	Counter
Dimensions	• application • hostname
Example	• occnp_ue_audit_notif_sent_to_amf_count_total{application="pcf_ueservice",hostname="pcf-pcf-ueservice-7988ccfbd9-zn8hs",} 1.0

Table 9-150 occnp_ue_audit_update_notify_session_found_total

Field	Details
Description	Total number of sessions found at AMF on sending update notify request (that is, when received 204 response from AMF)
Metric Type	Counter
Dimensions	• application • hostname
Example	• occnp_ue_audit_update_notify_session_found_total{application="pcf_ueservice",hostname="pcf-pcf-ueservice-58f8bd57d5-frggs",} 1.0

Table 9-151 occnp_ue_audit_update_notify_session_not_found_total

Field	Details
Description	Total number of sessions not found at AMF on sending update notify request (that is, received 404 response from AMF)
Metric Type	Counter
Dimensions	• application • hostname
Example	• occnp_ue_audit_update_notify_session_not_found_total{application="pcf_ueservice",hostname="pcf-pcf-ueservice-58f8bd57d5-frggs",} 1.0

Table 9-152 occnp_ue_audit_deleted_records_maxttl_reached_count_total

Field	Details
Description	Total number of records deleted as maxttl expired
Metric Type	Counter
Dimensions	• application • hostname
Example	• occnp_ue_audit_deleted_records_maxttl_reached_count_total{application="pcf_ueservice",hostname="pcf-pcf-ueservice-58f8bd57d5-frggs",} 1.0

Table 9-153 session_info_request_total

Field	Details
Description	Total number of sessions to be deleted as requested by the UE Policy Service
Metric Type	Counter
Dimensions	• application • cause • operation_type • session_type

Table 9-153 (Cont.) session_info_request_total

Field	Details
Example	• session_info_request_total{application="pcf_ueservice",cause="local_limiting_session",operation_type="delete",session_type="ue_session",} 3.0

Table 9-154 session_info_response_total

Field	Details
Description	Total number of sessions deleted as requested by the UE Policy Service
Metric Type	Counter
Dimensions	• application • cause • operation_type • response_code • session_type
Example	• session_info_response_total{application="pcf_ueservice",cause="local_limiting_session",operation_type="delete",response_code="5xx",session_type="ue_session",} 2.0

Table 9-155 occnp_ue_discovery_success_count_total

Field	Details
Description	Enables Policy to report how many discoveries based in AMF Region ID and AMF Set ID or GUAMI were successful
Metric Type	Counter
Dimensions	• discoveryType • setID • regionID • mcc • mnc • amfId
Example	• occnp_ue_discovery_success_count_total '{discoveryType="setIdRegionId", setID="2", regionID="1",mcc="",mnc="",amfId=""}',2 • occnp_ue_discovery_success_count_total '{discoveryType="guami",setID="",regionID="", mcc="450",mnc="05",amfId="010041"}',2

Table 9-156 occnp_ue_discovery_failure_count_total

Field	Details
Description	Enables Policy to report how many discoveries based in AMF Region ID and AMF Set ID or GUAMI failed

Table 9-156 (Cont.) occnp_ue_discovery_failure_count_total

Field	Details
Metric Type	Counter
Dimensions	• discoveryType • setID • regionID • mcc • mnc • amfId
Example	• occnp_ue_discovery_failure_count_total '{discoveryType="setIdRegionId", setID="1", regionID="2",mcc="",mnc="",amfId=""}',1

Table 9-157 occnp_http_ue_request_total

Field	Details
Description	Number of http UE requests
Metric Type	Counter
Dimensions	• application • hostname • operationType • servicenameNon3gpp
Example	• occnp_http_ue_request_total{application="pcf_ueservice",hostname="jiandong-pcf-ueservice-5878554dc7-wgvc7",operationType="bulwark_lock",servicenameNon3gpp="npcf-ueservice"}

Table 9-158 occnp_http_ue_response_total

Field	Details
Description	Number of http UE responses
Metric Type	Counter
Dimensions	• application • hostname • operationType • servicenameNon3gpp
Example	• occnp_http_ue_response_total{application="pcf_ueservice",hostname="ueservice-5878554dc7-wgvc7",operationType="bulwark_unlock", responseCode="204",servicenameNon3gpp="npcf-ueservice", retry=false}

Table 9-159 http_out_conn_request_total

Field	Details
Description	Total number of connection requests recieved
Metric Type	Counter

Table 9-159 (Cont.) http_out_conn_request_total

Field	Details
Dimensions	• Application • bindingId • bindingLevel • discoveryId • discoveryParameter • hostname • nfInstanceId • nfName • operationType • servicename3gpp
Example	• http_out_conn_request_total{application="pcf_ueservice",bindingId="set001.region48.amfset.5gc.mnc012.mcc345",bindingLevel="nf-set",discoveryId="set001.region48.amfset.5gc.mnc012.mcc345",discoveryParameter="target-nf-set-id",hostname="my-cnpolicy-pcf-ueservice-6696fd4bf4-6mf8v",nfInstanceId="13515195-c537-4645-9b97-96ec797f2222",nfName="nf1stub.harald-ns.svc",operationType="transfer",servicename3gpp="npcf-ue-policy-control",} 3.0

Table 9-160 http_out_conn_response_total

Field	Details
Description	Total number of connection response sent
Metric Type	Counter
Dimensions	• Application • hostname • nfInstanceId • nfName • operationType • servicename3gpp
Example	• http_out_conn_response_total{application="pcf_ueservice",hostname="my-cnpolicy-pcf-ueservice-7d5bc8d8d4-gsnr4",nfInstanceId="13515195-c537-4645-9b97-96ec797f2222",nfName="nf1stub.harald-ns.svc",operatione="2xx",servicename3gpp="npcf-ue-policy-control",} 3.0

Table 9-161 ocpm_ar_request_total

Field	Details
Description	Total number of ar requests recieved
Metric Type	Counter

Table 9-161 (Cont.) ocpm_ar_request_total

Field	Details
Dimensions	• Application • fqdn • hostname • scheme • servicename3gpp
Example	• ocpm_ar_request_total{application="pcf_ueservice",fqdn="nf22stub.harald-ns.svc",hostname="my-cnpolicy-pcf-ueservice-7d5bc8d8d4-gsnr4",scheme="http",servicename3gpp="npcf-ue-policy-control",} 1.0

Table 9-162 ocpm_ar_response_total

Field	Details
Description	Total number of ar response sent
Metric Type	Counter
Dimensions	• Application • fqdn • hostname • scheme • servicename3gpp
Example	• ocpm_ar_response_total{application="pcf_ueservice",fqdn="nf22stub.harald-ns.svc",hostname="my-cnpolicy-pcf-ueservice-7d5bc8d8d4-gsnr4",responseCode="4xx",scheme="http",servicename3gpp="npcf-ue-policy-control",} 1.0

Table 9-163 topic_version

Field	Details
Description	Describes the current applied version of a given topic (mentioned in dimension topic_name) into the pod.
Metric Type	Counter
Dimensions	• Service Name • Pod Name

Table 9-164 n1_message_policy_evaluation

Field	Details
Description	Metric that increments each time there was a policy evaluated for skip, abort and retransmit n1 notification message.
Metric Type	Counter

Table 9-164 (Cont.) n1_message_policy_evaluation

Field	Details
Dimensions	• action (action that has been taken skip/abort/ retransmit/retransmit_fragment) • message_type (MANAGE UE POLICY COMMAND REJECT/ N1N2MessageTransferFailure)

Table 9-165 occnp_db_overall_processing_time_seconds_sum

Field	Details
Description	This metric is incremented when PCF performs DELETE operation on an Association
Metric Type	Counter
Dimensions	• Method • Status • Table
Example	• occnp_db_overall_processing_time_seconds_ sum{Method="delete", Status="Success", Table="UePolicyAssociation", container="ue-service"}

Table 9-166 occnp_db_overall_processing_time_seconds_sum

Field	Details
Description	This metric is incremented when PCF performs DELETE operation on an Association
Metric Type	Counter
Dimensions	• Method • Status • Table
Example	• occnp_db_overall_processing_time_seconds_ sum{Method="delete", Status="Success", Table="UePolicyAssociation", container="ue-service"}

Table 9-167 occnp_ueservice_overall_processing_time_seconds_max

Field	Details
Description	This metric is incremented for UE service to calculate processing's duration of HTTP request arrived on rest controller. **Note**: occnp_ueservice_overall_processing_time_seconds_max metric replaces http_server_requests_seconds_max.
Metric Type	Counter

Table 9-167 (Cont.) occnp_ueservice_overall_processing_time_seconds_max

Field	Details
Dimensions	• Method • Status • uri • container
Example	• occnp_ueservice_overall_processing_time_seconds_max{method="DELETE", status="204", uri="/npcf-ue-policy-control/v1/policies/{polAssoId}", container="ue-service"}

Table 9-168 occnp_ueservice_overall_processing_time_seconds_sum

Field	Details
Description	This metric is incremented for UE service to calculate processing's duration of HTTP request arrived on rest controller. **Note**: occnp_ueservice_overall_processing_time_seconds_sum metric replaces http_server_requests_seconds_sum.
Metric Type	Counter
Dimensions	• Method • Status • uri • container
Example	• occnp_ueservice_overall_processing_time_seconds_sum{method="DELETE", status="204", uri="/npcf-ue-policy-control/v1/policies/{polAssoId}", container="ue-service"}

Table 9-169 occnp_ueservice_overall_processing_time_seconds_count

Field	Details
Description	This metric is incremented for UE service to calculate processing's duration of HTTP request arrived on rest controller. **Note**: occnp_ueservice_overall_processing_time_seconds_count metric replaces http_server_requests_seconds_count.
Metric Type	Counter
Dimensions	• Method • Status • uri • container
Example	• occnp_ueservice_overall_processing_time_seconds_count{method="DELETE", status="204", uri="/npcf-ue-policy-control/v1/policies/{polAssoId}", container="ue-service"}

Table 9-170 ue_n1_unsubscribe_request_total

Field	Details
Description	This metric is incremented while UE service sends request to NAS-AMF for N1 UNSUBSCRIBE.
Metric Type	Counter
Dimensions	• servingPlmn
Example	• ue_n1_unsubscribe_request_total{servingPlmn="450005"}

Table 9-171 ue_n1_unsubscribe_response_total

Field	Details
Description	This metric is incremented when NAS-AMF sends back HTTP response to UE service for N1 UNSUBSCRIBE.
Metric Type	Counter
Dimensions	• latency • responseCode • servingPlmn
Example	• ue_n1_unsubscribe_response_total{latency="3",servingPlmn="450005",responseCode="204"}

Table 9-172 ue_n1_unsubscribe_exception_total

Field	Details
Description	This metric is incremented in case of any exception occured while sending N1 UNSUBSCRIBE request to NAS-AMF.
Metric Type	Counter
Dimensions	• servingPlmn
Example	• ue_n1_unsubscribe_exception_total{servingPlmn="450005"}

Table 9-173 error_handler_in_total

Field	Details
Description	This metric is incremented on initialization of error handling.
Type	Counter
Dimensions	• application • applicationException • wrapped_exception
Example	occnp_error_handler_in_total{application="pcf_ueservice",application_exception="JavaException",status="400",wrapped_exception="ServiceException",} 6.0

Table 9-174 error_handler_exec_total

Field	Details
Description	This metric is incremented on error handling performed by framework.
Type	Counter
Dimensions	• application • applicationException • error_type • operation • origin • rule_name • source_interface • target_interface • wrapped_exception
Example	occnp_error_handler_exec_total{application="pcf_ueservice",application_exception="JavaException",error_type="INTERNAL",operation="UPDATE",origin="JAVA",rule_name="REJECT_WITH_ENHANCED_DETAIL",source_interface="AMF",status="411",target_interface="POLICY",wrapped_exception="ServiceException",} 1.0

Table 9-175 error_handler_out_total

Field	Details
Description	This metric is incremented on completion of error handling.
Type	Counter
Dimensions	• application • applicationException • error_resolved • wrapped_exception
Example	occnp_error_handler_out_total{application="pcf_ueservice",application_exception="JavaException",error_resolved="false",status="500",wrapped_exception="ServiceException",} 11.0

Table 9-176 ue_n1_subscribe_request_total

Field	Details
Description	This metric is incremented when PCF sends a request to AMF for N1 subscription.
Type	Counter
Dimensions	servingPlmn
Example	ue_n1_subscribe_request_total{application="pcf_ueservice",hostname="rel-pcf-ueservice-67d4fd6b49-sqz57",servingPlmn="330042",}

Table 9-177 ue_n1_subscribe_response_total

Field	Details
Description	This metric is incremented when PCF receives a response from AMF for N1 subscription request.
Type	Counter

Table 9-177 (Cont.) ue_n1_subscribe_response_total

Field	Details
Dimensions	• latency • responseCode • servingPlmn
Example	ue_n1_subscribe_response_total{application="pcf_ueservice",hostname="rel-pcf-ueservice-67d4fd6b49-sqz57",latency="5",responseCode="201",servingPlmn="330042",}

For more information about Dimensions, see CNC Policy Metrics.

9.11 User Service Metrics

The following table describes the User Service (User/UDR/CHF) metrics and respective dimensions:

Table 9-178 ocpm_userservice_inbound_count_total

Field	Details
Description	Total number of inbound request.
Metric Type	Counter
Dimensions	• operation_type • service_resource

Examples:

- ocpm_userservice_inbound_count_total{application="pcf_userservice",operation_type="post",service_resource="udr-service",} 0.0 ; Type-Counter
- ocpm_userservice_inbound_count_total{application="pcf_userservice",operation_type="get",service_resource="chf-service",} 0.0 ; Type-Counter
- ocpm_userservice_inbound_count_total{application="pcf_userservice",operation_type="get",service_resource="udr-service",} 0.0 ; Type-Counter
- ocpm_userservice_inbound_count_total{application="pcf_userservice",operation_type="notify",service_resource="chf-service",} 0.0 ; Type-Counter
- ocpm_userservice_inbound_count_total{application="pcf_userservice",operation_type="delete",service_resource="user-service",} 0.0 ; Type-Counter
- ocpm_userservice_inbound_count_total{application="pcf_userservice",operation_type="get",service_resource="user-service",} 0.0 ; Type-Counter
- ocpm_userservice_inbound_count_total{application="pcf_userservice",operation_type="notify",service_resource="udr-service",} 0.0 ; Type-Counter
- ocpm_userservice_inbound_count_total{application="pcf_userservice",operation_type="delete",service_resource="udr-service",} 0.0 ; Type-Counter
- ocpm_userservice_inbound_count_total{application="pcf_userservice",operation_type="terminate",service_resource="chf-service",} 0.0 ; Type-Counter
- ocpm_userservice_inbound_count_total{application="pcf_userservice",operation_type="delete",service_resource="chf-service",} 0.0 ; Type-Counter

- ocpm_userservice_inbound_count_total{application="pcf_userservice",operation_type="patch",service_resource="udr-service",} 0.0 ; Type-Counter
- ocpm_userservice_inbound_count_total{application="pcf_userservice",operation_type="put",service_resource="udr-service",} 0.0 ; Type-Counter

Table 9-179 ocpm_ar_setup_total

Field	Details
Description	Track number of registration request sent to alternate route service.
Metric Type	Counter
Dimensions	• anchorFqdn • application • scheme

Examples:

- ocpm_ar_setup_total{anchorFqdn="chfanchor.allsim.svc",application="pcf_userservice",scheme="http",} 1.0 ; Type-Counter

Table 9-180 ocpm_ar_request_total

Field	Details
Description	Tracks number of alternate routing attempts to producer or consumer on Initial, Subsequent, or Notification messages.
Metric Type	Counter
Dimensions	• application • Fqdn • scheme

Examples:

- ocpm_ar_request_total{application="pcf_userservice",fqdn="udmanchor.allsim.svc",scheme="http",} 1.0 ; Type-Counter

Table 9-181 ocpm_ar_response_total

Field	Details
Description	Tracks response on number of alternate routing attempts to producer or consumer on Initial, Subsequent, or Notification messages.
Metric Type	Counter
Dimensions	• application • Fqdn • scheme

Examples:

- ocpm_ar_response_total{application="pcf_userservice",fqdn="udmanchor.allsim.svc",responseCode="2xx",scheme="http",} 1.0 ; Type-Counter

Table 9-182 ocpm_ar_lookup_request_total

Field	Details
Description	Track number of time alternate route lookup was done for a given scheme+FQDN.
Metric Type	Counter
Dimensions	• anchorFqdn • application • scheme

Examples:

- ocpm_ar_lookup_request_total{anchorFqdn="udmanchor.allsim.svc",application="pcf_user service",scheme="http",} 1.0 ; Type-Counter

Table 9-183 ocpm_ar_lookup_response_total

Field	Details
Description	Tracks response codes for request to "alternateRoute" service.
Metric Type	Counter
Dimensions	• anchorFqdn • application • responseCode • scheme

Examples:

- ocpm_ar_lookup_response_total{anchorFqdn="udmanchor.allsim.svc",application="pcf_us erservice",responseCode="2xx",scheme="http",} 1.0 ; Type-Counter

Table 9-184 ocpm_udr_tracking_request_total

Field	Details
Description	Total number of UDR tracking request.
Metric Type	Counter
Dimensions	• operation_type • nf_instance_id • nf_name • servicename_3gpp • service_resource • service_version • service_subresource • binding_level • binding_id • discovery_level • discovery_id

Examples:

- ocpm_udr_tracking_request_total{HostName="",application="pcf_userservice",nf_instance _id="fe7d992b-0541-4c7d-ab84-

c6d70b1babc1",operation_type="get",service_resource="policy-data",service_subresource="ue-policy-set",service_version="v1",servicename_3gpp="nudr-dr",} 0.0 ; Type-Counter

Table 9-185 ocpm_chf_tracking_request_total

Field	Details
Description	Total number of CHF tracking requests.
Metric Type	Counter
Dimensions	• operation_type • nf_instance_id • nf_name • servicename_3gpp • service_resource ["subscriptions"] • service_version ["v1,v1"] • binding_level • binding_id • discovery_level • discovery_id

Table 9-186 ocpm_udr_tracking_response_total

Field	Details
Description	Total number of UDR tracking responses.
Metric Type	Counter
Dimensions	• operation_type • nf_instance_id • nf_name • servicename_3gpp= ["nudr-dr"] • service_resource ["policy-data"] • service_version ["v1,v2"] • service_subresource [am-data, sm-data, ue-policy-set, subs-to-notify] • response_code

Examples:

- ocpm_udr_tracking_response_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-c6d70b1babc1",nf_name="udr.oracle.com",operation_type="subscribe",response_code="2xx",service_resource="policy-data",service_subresource="",service_version="v1",servicename_3gpp="nudr-dr",} 0.0 ; Type-Counter

Table 9-187 ocpm_udr_tracking_request_timeout_total

Field	Details
Description	Total number of UDR request timeouts.
Metric Type	Counter

Table 9-187 (Cont.) ocpm_udr_tracking_request_timeout_total

Field	Details
Dimensions	• Hostname • application • nf_instance_id • nf_name • operation_type • service_resource • service_subresource • service_version • servicename_3gpp

Examples:

- ocpm_udr_tracking_request_timeout_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-c6d70b1babc1",nf_name="udr.oracle.com",operation_type="patch",service_resource="policy-data",service_subresource="subs-to-notify",service_version="v1",servicename_3gpp="nudr-dr",} 0.0; Type-Counter

Table 9-188 ocpm_chf_tracking_response_total

Field	Details
Description	Total number of CHF tracking responses.
Metric Type	Counter
Dimensions	• operation_type • nf_instance_id • nf_name • servicename_3gpp= ["nchf-spendinglimitcontrol"] • service_resource ["subscriptions"] • service_version ["v1"] • response_code

Table 9-189 ocpm_chf_tracking_request_timeout_total

Field	Details
Description	Total number of CHF request timeouts.
Metric Type	
Dimensions	• Hostname • application • nf_instance_id • nf_name • operation_type • service_resource • service_subresource • service_version • servicename_3gpp

Table 9-190 topic_version

Field	Details
Description	Describes the current applied version of a given topic (mentioned in dimension topic_name) into the pod.
Metric Type	Gauge
Dimensions	• Service Name • Pod Name

Table 9-191 occnp_policy_userservice_overall_processing_time_seconds

Field	Details
Description	User service overall processing time. **Note**: Default name used when there is no custom name available is spring_data_repository_invocations.
Type	Summary
Dimension	• application • error • exception • hostname • method • outcome • status • uri
Example	

Table 9-192 occnp_policy_userservice_overall_processing_time_seconds_count

Field	Details
Description	User service overall processing time.
Type	Counter
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Table 9-193 occnp_policy_userservice_overall_processing_time_seconds_max

Field	Details
Description	User service overall processing time.
Type	Gauge

Table 9-193 (Cont.) occnp_policy_userservice_overall_processing_time_seconds_max

Field	Details
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Table 9-194 occnp_policy_userservice_overall_processing_time_seconds_sum

Field	Details
Description	User service overall processing time.
Type	Counter
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Table 9-195 occnp_udr_nf_cookie_enabled_total

Field	Details
Description	This metric is incremented when cached NF profile is used and the feature is enabled.
Metric Type	Counter
Dimensions	• HostName • application • hostname • nf_name • operation_type • service_resource • service_subresource • service_version • servicename_3gpp

Table 9-195 (Cont.) occnp_udr_nf_cookie_enabled_total

Field	Details
Example	occnp_udr_nf_profile_cookie_enabled_total{HostName="",application="pcf_userservice",hostname="pcf-occnp-udr-connector-5dc478c84c-78k9p",nf_instance_id="fe7d992b-0541-4c7d-ab84-555550000000",nf_name="nf1stub.s-galeleo.svc",operation_type="GET",service_resource="policy-data",service_subresource="ue-policy-set",service_version="v1",servicename_3gpp="nudr-dr",} 3.0

Examples:

- occnp_udr_nf_profile_cookie_enabled_total{HostName="",application="pcf_userservice",hostname="pcf-occnp-udr-connector-5dc478c84c-78k9p",nf_instance_id="fe7d992b-0541-4c7d-ab84-555550000000",nf_name="nf1stub.s-galeleo.svc",operation_type="GET",service_resource="policy-data",service_subresource="ue-policy-set",service_version="v1",servicename_3gpp="nudr-dr",} 3.0

Table 9-196 occnp_udr_use_related_resource

Field	Details
Description	This metric is incremented when a related resource is used and feature is enabled.
Metric Type	Counter
Dimensions	• HostName • application • hostname • nf_name • operation_type • service_resource • service_subresource • service_version • servicename_3gpp

Table 9-196 (Cont.) occnp_udr_use_related_resource

Field	Details
Example	occnp_udr_use_related_resource_total{HostName="",application="pcf_userservice",hostname="pcf-occnp-udr-connector-5dc478c84c-78k9p",nf_instance_id="fe7d992b-0541-4c7d-ab84-555550000000",nf_name="nf1stub.s-galeleo.svc",operation_type="GET",service_resource="policy-data",service_subresource="ue-policy-set",service_version="v1",servicename_3gpp="NUDR_DR",} 1.0

Table 9-197 error_handler_in_total

Field	Details
Description	This metric is incremented on initialization of error handling.
Type	Counter
Dimensions	• application • applicationException • wrapped_exception
Example	occnp_error_handler_in_total{application="pcf_userservice",application_exception="JavaException",status="502",wrapped_exception="ServiceException",} 4.0

Table 9-198 error_handler_exec_total

Field	Details
Description	This metric is incremented on error handling performed by framework.
Type	Counter
Dimensions	• application • applicationException • error_type • operation • origin • rule_name • source_interface • target_interface • wrapped_exception
Example	occnp_error_handler_exec_total{application="pcf_userservice",application_exception="JavaException",error_type="INTERNAL",operation="NOTIFY",origin="JAVA",rule_name="REJECT_WITH_ENHANCED_DETAIL",source_interface="UDR",status="502",target_interface="POLICY",wrapped_exception="ServiceException",} 4.0

Table 9-199 error_handler_out_total

Field	Details
Description	This metric is incremented on completion of error handling.
Type	Counter
Dimensions	• application • applicationException • error_resolved • wrapped_exception
Example	occnp_error_handler_out_total{application="pcf_userservice",application_exception="JavaException",error_resolved="true",status="400",wrapped_exception="ServiceException",} 4.0

For more information about Dimensions, see CNC Policy Metrics.

9.12 Diameter Connector Service Metrics

The following table describes the Diameter Connector Service metrics and respective dimensions:

Table 9-200 ocpm_ingress_request_total

Field	Details
Description	Total number of ingress requests received.
Metric Type	Counter
Dimensions	• operation_type • dnn • nf_instance_id • servicename_3gpp

Examples:

- ocpm_ingress_request_total{apn="",nf_instance_id="AF.oracle.com",operation_type="create",servicename_3gpp="rx",} 5.0 ; Type-Counter

Table 9-201 ocpm_ingress_response_total

Field	Details
Description	Total number of ingress responses.
Metric Type	Counter
Dimensions	• operation_type • nf_instance_id • dnn • servicename_3gpp • response_code

Examples:

- ocpm_ingress_response_total{apn="",nf_instance_id="ocpcf",operation_type="create",response_code="2xxx",servicename_3gpp="rx",} 2.0 ; Type-Counter

Table 9-202 ocpm_egress_request_total

Field	Details
Description	Total number of egress requests received.
Metric Type	Counter
Dimensions	• operation_type • dnn • nf_instance_id • servicename_3gpp

Examples:

- ocpm_egress_request_total{nf_instance_id="AF.oracle.com",operation_type="update_notify",servicename_3gpp="rx",} 1.0 ; Type-Counter

Table 9-203 ocpm_egress_response_total

Field	Details
Description	Total number of egress responses.
Metric Type	Counter
Dimensions	• operation_type • nf_instance_id • dnn • servicename_3gpp • response_code

Example:

- ocpm_egress_response_total{latency="3",nf_instance_id="AF.oracle.com",operation_type="update_notify",response_code="2xxx",servicename_3gpp="rx",} 1.0 ; Type-Counter

Table 9-204 topic_version

Field	Details
Description	Describes the current applied version of a given topic (mentioned in dimension topic_name) into the pod.
Metric Type	Gauge
Dimensions	• Service Name • Pod Name

Table 9-205 occnp_diam_connector_overall_processing_time_seconds

Field	Details
Description	Diameter Connector service overall processing time.
Type	Summary

Table 9-205 (Cont.) occnp_diam_connector_overall_processing_time_seconds

Field	Details
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Table 9-206 occnp_diam_connector_overall_processing_time_seconds_max

Field	Details
Description	Diameter Connector service overall processing time. **Note**: Default name used when there is no custom name available is spring_data_repository_invocations.
Type	Gauge
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Table 9-207 occnp_diam_connector_overall_processing_time_seconds_count

Field	Details
Description	Diameter Connector service overall processing time.
Type	Counter
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Table 9-208 occnp_diam_connector_overall_processing_time_seconds_sum

Field	Details
Description	Diameter Connector service overall processing time.
Type	Counter

Table 9-208 (Cont.) occnp_diam_connector_overall_processing_time_seconds_sum

Field	Details
Dimension	• application • error • exception • hostname • method • outcome • status • uri

For more information about Dimensions, see CNC Policy Metrics.

9.13 Diameter Gateway Metrics

The following table describes the Diameter Gateway metrics and respective dimensions:

Table 9-209 occnp_diam_conn_network

Field	Details
Description	Tracks active diameter connections with network peers.
Type	Gauge
Dimension	• destHost • destRealm • direction • origHost • origRealm
Example	occnp_diam_conn_network{destHost="",destRealm="",direction="in",origHost="udr.oracle.com",origRealm="oracle.com",} 1.0 ; Type-Gauge

Table 9-210 occnp_diam_conn_app_network

Field	Details
Description	Tracks active diameter connections with network peers that support given application ID.
Type	Gauge
Dimension	• appId • applicationName • destHost • destRealm • origHost • origRealm
Examples	• occnp_diam_conn_app_network{appId="16777217",applicationName="Sh",destHost="",destRealm="",origHost="udr.oracle.com",origRealm="oracle.com",} 1.0 ; Type-Gauge **Note**: Sh interface is not supported for Converged Policy mode of deployment. • occnp_diam_conn_app_network{appId="16777217",applicationName="Sh",destHost="",destRealm="",origHost="udr.oracle.com",origRealm="oracle.com",} 1.0 ; Type-Gauge

Example:

Table 9-211 occnp_diam_request_network_total

Field	Details
Description	Tracks total number of request messages of given command code to or from network
Type	Counter
Dimension	• appId • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm • retry • retryReason
Example	occnp_diam_request_network_total{appId="16777217",cmdCode="308",destHost="",destRealm="oracle.com",direction="out",msgType="SNR",origHost="pcrf-core",origRealm="oracle.com",} 2.0 ; Type-Counter

Table 9-212 occnp_diam_response_network_total

Field	Details
Description	Tracks total number of answer messages of given command code to or from network.
Type	Counter
Dimension	• appId • cmdCode • reqDestHost • reqDestRealm • direction • msgType • reqOrigHost • reqOrigRealm • responseCode
Example	occnp_diam_response_network_total{appId="16777217",cmdCode="308",reqDestHost="udr.oracle.com",reqDestRealm="oracle.com",direction="in",msgType="SNA",reqOrigHost="pcrf-core",reqOrigRealm="oracle.com",responseCode="2001",} 2.0 ; Type-Counter

Table 9-213 occnp_diam_conn_local

Field	Details
Description	Tracks active diameter connections with local services.
Type	Gauge
Dimension	• destHost • destRealm • direction • origHost • origRealm
Example	occnp_diam_conn_local{destHost="",destRealm="",direction="out",origHost="pcrf-core",origRealm="oracle.com",} 1.0 ; Type-Gauge

Table 9-214 occnp_diam_conn_app_local

Field	Details
Description	Tracks active diameter connections with local services that support given application id. Increment this metric for each application Id supported over connection, such as intersection of application ID in CER and CEA.
Type	Gauge
Dimension	• appId • appType • applicationName • destHost • destRealm • origHost • origRealm
Example	occnp_diam_conn_app_local{appId="16777294,16777222,16777266,16777302,16777236,16777237,16777303,16777267,16777322,16777229,16777238,16777217",appType="pcrf",applicationName="",destHost="",destRealm="",origHost="pcrf-core",origRealm="oracle.com",} 1.0 ; Type-Gauge

Table 9-215 occnp_diam_request_local_total

Field	Details
Description	Tracks total number of request messages of given command code to or from local service
Type	Counter
Dimension	• appId • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm • reqDestHost • reqDestRealm • reqOrigHost • reqDestHost • retry • retryReason
Examples	• occnp_diam_request_local_total{appId="16777303",appType="",cmdCode="8388637",destHost="pcrf-core",destRealm="oracle.com",direction="out",msgType="TSR",origHost="local-pcrf",origRealm=local.com",reqDestHost="tdf.oracle.com",reqDestRealm="tdf.oracle.com",reqOrigHost="pcrf-core",reqOrigRealm="oracle.com",} 1.0 1658786894876 • occnp_diam_request_local_total{appId="16777217",cmdCode="308",destHost="",destRealm="oracle.com",direction="in",msgType="SNR",origHost="pcrf-core",origRealm="oracle.com",} 2.0 ; Type-Counter • occnp_diam_request_local_total{appId="16777217",cmdCode="306",destHost="",destRealm="oracle.com",direction="in",msgType="UDR",origHost="pcrf-core",origRealm="oracle.com",} 5.0 ; Type-Counter

Table 9-216 occnp_diam_response_local_total

Field	Details
Description	Tracks total number of answer messages of given command code to or from local service.
Type	Counter
Dimension	• appId • cmdCode • reqDestHost • reqDestRealm • direction • msgType • reqOrigHost • reqOrigRealm • responseCode
Examples	• occnp_diam_response_local_total{appId="16777217",cmdCode="308",reqDestHost="udr.oracle.com",reqDestRealm="oracle.com",direction="out",msgType="SNA",reqOrigHost="pcrf-core",rqOrigRealm="oracle.com",responseCode="2001",} 2.0 ; Type-Counter • occnp_diam_response_local_total{appId="16777217",cmdCode="306",reqDestHost="udr.oracle.com",reqDestRealm="oracle.com",direction="out",msgType="UDA",reqOrigHost="pcrf-core",reqOrigRealm="oracle.com",responseCode="2001",} 3.0 ; Type-Counter

Table 9-217 occnp_diam_request_inter_total

Field	Details
Description	Tracks total number of request messages of given command code to or from host services
Type	Counter
Dimension	• appId • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm • retry • retryReason

Table 9-218 occnp_http_out_conn_request_total

Field	Details
Description	Total number of http connection requests.
Type	Counter
Dimension	• operationType • serviceResource • serviceVersion • servicename3gpp
Example	occnp_http_out_conn_request_total{operationType="read",serviceResource="Binding",service_version="v1",servicename3gpp="DiameterGateway",} 1.0 ; Type-Counter

Table 9-219 occnp_http_out_conn_response_total

Field	Details
Description	Total number of http connection responses.
Type	Counter
Dimension	• latency • operationType • serviceResource • serviceVersion • servicename3gpp
Example	occnp_http_out_conn_response_total{latency="9",operationType="read",responseCode="2xx",serviceResource="Binding",serviceVersion="v1",servicename3gpp="DiameterGateway",} 1.0 ; Type-Counter

Table 9-220 diam_overload_msg_reject_total

Field	Details
Description	Total number of messages rejected due to overload control.
Type	Counter
Dimension	• priority • response_code • destHost • destRealm • origHost • origRealm • appId • cmdCode • msgType

Table 9-221 diam_controlled_shutdown_message_reject_total

Field	Details
Description	Indicates failure count because of forced shutdown feature.
Type	Counter
Dimension	• state • response_code • destHost • destRealm • origHost • origRealm • appId • cmdCode • msgType • direction
Example	diam_controlled_shutdown_message_reject_total{msgType="Gx_CCR_I",appId="16777238",cmdCode="272",destHost="",destRealm="oracle.com",operationalState="PARTIAL_SHUTDOWN",origHost="pgw.oracle.com",origRealm="test.example.com",responseCode="5012",} 1.0

Table 9-222 occnp_diam_overload_message_reject_total

Field	Details
Description	Indicates the number of messages rejected due to overload.
Type	Counter
Dimension	• priority • response_code • destHost • destRealm • origHost • origRealm • appId • cmdCode • msgType • direction
Example	occnp_diam_overload_message_reject_total{appId="16777236",cmdCode="265",destHost="",destRealm="oracle.com",msgType="request",origHost="diamcliaf.oracle.com",origRealm="test.example.com",priority="12",responseCode="3004",} 1.0

Table 9-223 occnp_diam_congestion_message_reject_total

Field	Details
Description	Indicates the number of messages rejected due to congestion.
Type	Counter
Dimension	• appId • cmdCode • destHost • destRealm • msgType • origHost • origRealm • responseCode
Examples	• occnp_diam_congestion_message_reject_total{appId="16777238",cmdCode="272",destHost="pcrf-core",destRealm="oracle.com",msgType="request",origHost="diamclipgw.oracle.com",origRealm="oracle.com",priority="11",responseCode="3004",} 38.0; Type-Counter • occnp_diam_congestion_message_reject_total{appId="16777238",cmdCode="272",destHost="pcrf-core",destRealm="oracle.com",msgType="request",origHost="diamclipgw.oracle.com",origRealm="oracle.com",priority="10",responseCode="3004",} 61.0; Type-Counter • occnp_diam_congestion_message_reject_total{appId="16777238",cmdCode="272",destHost="pcrf-core",destRealm="oracle.com",msgType="request",origHost="diamclipgw.oracle.com",origRealm="oracle.com",priority="11",responseCode="3002",} 8.0 Type-Counter

Table 9-224 topic_version

Field	Details
Description	Describes the current applied version of a given topic (mentioned in dimension topic_name) into the pod.

Table 9-224 (Cont.) topic_version

Field	Details
Type	Gauge
Dimension	• Service Name • Pod Name

Table 9-225 occnp_pod_resource_congestion_state

Field	Details
Description	Indicates the pod congestion state.
Type	Gauge
Dimension	type
Examples	• occnp_pod_congestion_state 1.0; Type-Gauge • occnp_pod_resource_congestion_state{type="memory",} 1.0; Type-Gauge • occnp_pod_resource_congestion_state{type="cpu",} 0.0; Type-Gauge • occnp_pod_resource_congestion_state{type="queue",} 0.0; Type-Gauge • occnp_pod_resource_stress{type="memory",} 41.0; Type-Gauge • occnp_pod_resource_stress{type="cpu",} 0.0; Type-Gauge • occnp_pod_resource_stress{type="queue",} 0.0; Type-Gauge

Table 9-226 occnp_diameter_outstanding_msg_count

Field	Details
Description	Provides an instantaneous number of outstanding diameter messages in the queue.
Type	Gauge
Dimension	NA
Examples	occnp_diameter_outstanding_msg_count 10.0

For more information about Dimensions, seeCNC Policy Metrics.

9.14 Policy DS Metrics

The following table describes the Policy DS Service metrics and respective dimensions:

Table 9-227 client_request_total

Field	Details
Description	Total number of client requests.
Type	Counter
Dimension	• operation • task
Example	client_request_total{application="policyds",operation="SEARCH",workflow="LDAP",} 1.0 ; Type-Counter

Table 9-228 client_response_total

Field	Details
Description	Total number of responses from the client.
Type	Counter
Dimension	• operation • task • response
Example	client_response_total{application="policyds",operation="SEARCH",response="200",workflow="LDAP",} 1.0 ; Type-Counter

Table 9-229 server_request_total

Field	Details
Description	Metric that increments its value each time a request is being sent from PDS to another service. For example: For communication with CHF task dimension is "CHF".
Type	Counter
Dimension	• exported_application • hostname • method • task • vendorId
Examples	• server_request_total{exported_application="policyds",method="SEARCH",task="UDR",} 1.0 ; Type-Counter • server_request_total{exported_application="policyds",method="GET",task="LDAP",} 1.0 ; Type-Counter • server_request_total{exported_application="policyds",method="INSERT",task="PRE",} 1.0 ;Type-Counter

Table 9-230 server_response_total

Field	Details
Description	Metric that increments its value each time a response is being received from a request that PDS sent to another service. For example: For communication with CHF task dimension is "CHF". Response dimension specifies the response of the request (200, 500, 400, etc)
Type	Counter
Dimension	• exported_application • hostname • method • response • task • vendorId
Example	server_response_total{exported_application="policyds",method="POST",response="200",} 1.0 ; Type-Counter

Table 9-231 remove_contextInfo_total

Field	Details
Description	Total number of context infos deleted once the column length has reached the configured limit, that is, 4000 characters. **Note**: To modify the default value, update the value for the variable `CONTEXT_INFO_COLUMN_LIMIT` when deploying CNC Policy.
Type	Counter
Dimension	• application • cause • contextOwner • dnnSlice • sourceType • vendorId

Table 9-232 revalidation_request

Field	Details
Description	This metric is incremented when the revalidation is required for a datasource. It is also incremented when revalidation for a subscription failed with 404 but reattempt is done for the request.
Type	Counter
Dimension	• tagName • task • method • revalidationStatus • vendorId
Examples	• revalidation_request_total{application="policyds",hostname="plcy-ocpm-policyds-567d754bbd-7cwrt",method="INSERT",revalidation_status="REVALIDATION",task="UDR_SUBS",vendorId="Oracle",} 9.0 • revalidation_request_total{application="policyds",hostname="plcy-ocpm-policyds-567d754bbd-7cwrt",method="SEARCH",revalidation_status="REVALIDATION",task="CHF",vendorId="Oracle",} 11.0 • revalidation_request_total{application="policyds",hostname="plcy-ocpm-policyds-567d754bbd-7cwrt",method="SEARCH",revalidation_status="REVALIDATION_REATTEMPT",task="CHF",vendorId="Oracle",} 4.0

Table 9-233 revalidation_response

Field	Details
Description	This metric is incremented when any of the following responses are received for revalidation request: • When CHF or UDR query is successful returning 200 OK and revalidation required is also true. • CHF revalidation failed with an error other than 404 but reattempt is not required, thereby resulting in data removal. • CHF query is unsuccessful returning 404, and revalidation failed but reattempt is not to be done, thereby resulting in data removal. • UDR subscription query revalidation reattempt failed, so subscription information is removed for UDR datasource. • Revalidation request for UDR GET failed. **Note**: You can identify the exact cause with the `revalidationStatus` dimension of the metric.
Type	Counter
Dimension	• tagName • task • method • revalidationStatus • vendorId
Examples	• revalidation_response_total{application="policyds",hostname="plcy-ocpm-policyds-567d754bbd-7cwrt",method="SEARCH",revalidation_status="REVALIDATION_SUCCESS",task="CHF",vendorId="Oracle",} 9.0 • revalidation_response_total{application="policyds",hostname="plcy-ocpm-policyds-567d754bbd-7cwrt",method="INSERT",revalidation_status="REVALIDATION_REATTEMPT_DELETE_SUBSCRIPTION",task="UDR_SUBS",vendorId="Oracle",} 1.0 • revalidation_response_total{application="policyds",hostname="plcy-ocpm-policyds-567d754bbd-7cwrt",method="SEARCH",revalidation_status="REVALIDATION_REATTEMPT_DELETE_ENTITY",task="UDR",vendorId="Oracle",} 2.0

Table 9-234 topic_version

Field	Details
Description	Describes the current applied version of a given topic (mentioned in dimension topic_name) into the pod.
Type	Gauge
Dimension	• Service Name • Pod Name

Table 9-235 error_handler_in_total

Field	Details
Description	This metric is incremented on initialization of error handling.
Type	Counter
Dimension	• application • application_exception • wrapped_exception
Example	error_handler_in_total{application="policyds",application_exception="HttpException",wrapped_exception="TimeoutException",} 1.0

Table 9-236 error_handler_exec_total

Field	Details
Description	This metric is incremented on error handling performed by framework.
Type	Counter
Dimension	• application • application_exception • error_type • operation • origin • rule_name • source_interface • target • wrapped_exception
Example	error_handler_exec_total{application="policyds",application_exception="HttpException",error_type="INTERNAL",operation="LOOKUP",origin="HTTP",rule_name="HTTP_REQUEST",source_interface="POLICY",status="502",target_interface="BSF",wrapped_exception="BadGateway",} 1.0

Table 9-237 error_handler_out_total

Field	Details
Description	This metric is incremented on completion of error handling.
Type	Counter
Dimension	• application • application_exception • error_resolved • wrapped_exception

Table 9-238 occnp_policyds_overall_processing_time_seconds

Field	Details
Description	Policy DS service overall processing time. **Note**: Default name used when there is no custom name available is spring_data_repository_invocations.
Type	Summary
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Table 9-239 occnp_policyds_overall_processing_time_seconds_max

Field	Details
Description	Diameter Connector service overall processing time.

Table 9-239 (Cont.) occnp_policyds_overall_processing_time_seconds_max

Field	Details
Type	Gauge
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Table 9-240 occnp_policyds_overall_processing_time_seconds_count

Field	Details
Description	Diameter Connector service overall processing time.
Type	Counter
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Table 9-241 occnp_policyds_overall_processing_time_seconds_sum

Field	Details
Description	Diameter Connector service overall processing time.
Type	Counter
Dimension	• application • error • exception • hostname • method • outcome • status • uri
Examples	error_handler_out_total{application="policyds",application_exception="HttpException",error_resolved="true",wrapped_exception="BadRequest",} 1.0

Table 9-242 occnp_nf_cookie_forwarded_total

Field	Details
Description	This metric is incremented every time a NF profile cookie is forwarded from PDS.
Type	Counter

Table 9-242 (Cont.) occnp_nf_cookie_forwarded_total

Field	Details
Dimension	• application • task • vendorId
Example	occnp_nf_cookie_forwarded_total{application="policyds,"task="UDR",vendorId="Oracle"} 1.0 ;

Table 9-243 occnp_nf_cookie_recieved_total

Field	Details
Description	This metric is incremented every time a NF profile cookie is recieved from PDS.
Type	Counter
Dimension	• application • task • vendorId
Example	occnp_nf_cookie_recieved_total{application="policyds,task="UDR",vendorId="Oracle"} 1.0 ;

Table 9-244 spring_data_repository_invocations_seconds_max

Field	Details
Description	This metric is used to track the maximum execution time of Spring Data repository invocations. It is a useful metric for monitoring the performance of Spring Data repositories and identifying slow-running queries.
Type	Gauge

Table 9-244 (Cont.) spring_data_repository_invocations_seconds_max

Field	Details
Dimension	• app_kubernetes_io_instance • app_kubernetes_io_managed_by • app_kubernetes_io_name • app_kubernetes_io_part_of • app_kubernetes_io_version • application • container • endpoint • engVersion • exception • helm_sh_chart • instance • io_kompose_service • job • method • microservice • mktgVersion • namespace • pod • pod_template_hash • repository • security_istio_io_tlsMode • service_istio_io_canonical_name • service_istio_io_canonical_revision • state • vendor
Example	spring_data_repository_invocations_seconds_max{app_kubernetes_io_instance="ocbsf", app_kubernetes_io_managed_by="Helm", app_kubernetes_io_name="audit-service", app_kubernetes_io_part_of="ocbsf", app_kubernetes_io_version="1.0.0.0", application="ocbsf", container="audit-service", endpoint="cnc-metrics", engVersion="24.1.4", exception="None", helm_sh_chart="audit-service-24.1.4", instance="10.233.127.211:9000", io_kompose_service="ocbsf-ocpm-audit-service", job="occne-infra/occne-nf-cnc-podmonitor", method="getQueuedTablesToAudit", microservice="ocbsf-audit", mktgVersion="1.0.0.0", namespace="ocbsf", pod="ocbsf-ocpm-audit-service-6b6f85f7b9-nmzr4", pod_template_hash="6b6f85f7b9", repository="AuditSchedulerRepository", security_istio_io_tlsMode="istio", service_istio_io_canonical_name="audit-service", service_istio_io_canonical_revision="1.0.0.0", state="SUCCESS", vendor="Oracle"}

Table 9-245 spring_data_repository_invocations_seconds_sum

Field	Details
Description	The seconds_sum metric for Spring Data repository invocations is a measure of the total time spent executing repository methods.
Type	Gauge

Table 9-245 (Cont.) spring_data_repository_invocations_seconds_sum

Field	Details
Dimension	• app_kubernetes_io_instance • app_kubernetes_io_managed_by • app_kubernetes_io_name • app_kubernetes_io_part_of • app_kubernetes_io_version • application • container • endpoint • engVersion • exception • helm_sh_chart • instance • io_kompose_service • job • method • microservice • mktgVersion • namespace • pod • pod_template_hash • repository • security_istio_io_tlsMode • service_istio_io_canonical_name • service_istio_io_canonical_revision • state • vendor
Example	spring_data_repository_invocations_seconds_sum{app_kubernetes_io_instance="ocbsf", app_kubernetes_io_managed_by="Helm", app_kubernetes_io_name="audit-service", app_kubernetes_io_part_of="ocbsf", app_kubernetes_io_version="1.0.0.0", application="ocbsf", container="audit-service", endpoint="cnc-metrics", engVersion="24.1.4", exception="None", helm_sh_chart="audit-service-24.1.4", instance="10.233.84.88:9000", io_kompose_service="ocbsf-ocpm-audit-service", job="occne-infra/occne-nf-cnc-podmonitor", method="getQueuedTablesToAudit", microservice="ocbsf-audit", mktgVersion="1.0.0.0", namespace="ocbsf", pod="ocbsf-ocpm-audit-service-6b6f85f7b9-nshv9", pod_template_hash="6b6f85f7b9", repository="AuditSchedulerRepository", security_istio_io_tlsMode="istio", service_istio_io_canonical_name="audit-service", service_istio_io_canonical_revision="1.0.0.0", state="SUCCESS", vendor="Oracle"}

Table 9-246 spring.data.jpa.repositories.seconds_count

Field	Details
Description	This metric measures the total time spent in Spring Data JPA repository invocations. It is a counter metric, which means that it increments each time a repository method is invoked. The metric is reported in seconds.
Type	Counter

Table 9-246 (Cont.) spring.data.jpa.repositories.seconds_count

Field	Details
Dimension	• app_kubernetes_io_instance • app_kubernetes_io_managed_by • app_kubernetes_io_name • app_kubernetes_io_part_of • app_kubernetes_io_version • application • container • endpoint • engVersion • exception • helm_sh_chart • instance • io_kompose_service • job • method • microservice • mktgVersion • namespace • pod • service • pod_template_hash • repository • security_istio_io_tlsMode • service_istio_io_canonical_name • service_istio_io_canonical_revision • state • vendor
Example	spring_data_repository_invocations_seconds_count{app_kubernetes_io_instance="ocbsf", app_kubernetes_io_managed_by="Helm", app_kubernetes_io_name="audit-service", app_kubernetes_io_part_of="ocbsf", app_kubernetes_io_version="1.0.0.0", application="ocbsf", container="audit-service", endpoint="cnc-metrics", engVersion="24.1.4", exception="None", helm_sh_chart="audit-service-24.1.4", instance="10.233.127.211:9000", io_kompose_service="ocbsf-ocpm-audit-service", job="occne-infra/occne-nf-cnc-podmonitor", method="findAll", microservice="ocbsf-audit", mktgVersion="1.0.0.0", namespace="ocbsf", pod="ocbsf-ocpm-audit-service-6b6f85f7b9-nmzr4", pod_template_hash="6b6f85f7b9", repository="AuditRegistrationsRepository", security_istio_io_tlsMode="istio", service_istio_io_canonical_name="audit-service", service_istio_io_canonical_revision="1.0.0.0", state="SUCCESS", vendor="Oracle"}

Table 9-247 create_context_on_failure_response

Field	Details
Description	This metrics increments on receiving failure response from its respective data source and that has createContextOnFailure set to true in the request.
Metric Type	Counter
Dimensions	• task • method • sourceType

Table 9-247 (Cont.) create_context_on_failure_response

Field	Details
Example	

Table 9-248 error_handler_in_total

Field	Details
Description	This metric is incremented on initialization of error handling.
Type	Counter
Dimensions	• application • applicationException • wrapped_exception
Example	error_handler_in_total{application="policyds",application_exception="HttpException",status="404",wrapped_exception="NotFound",} 26.0

Table 9-249 error_handler_exec_total

Field	Details
Description	This metric is incremented on error handling performed by framework.
Type	Counter
Dimensions	• application • applicationException • error_type • operation • origin • rule_name • source_interface • target_interface • wrapped_exception
Example	error_handler_exec_total{application="policyds",application_exception="HttpException",error_type="INTERNAL",operation="LOOKUP",origin="HTTP",rule_name="DELETE_OR_UPDATE_UDR",source_interface="POLICY",status="500",target_interface="UDR",wrapped_exception="InternalServerError",} 2.0

Table 9-250 error_handler_out_total

Field	Details
Description	This metric is incremented on completion of error handling.
Type	Counter
Dimensions	• application • applicationException • error_resolved • wrapped_exception
Example	error_handler_out_total{application="policyds",application_exception="HttpException",error_resolved="false",status="404",wrapped_exception="NotFound",} 25.0

For more information about Dimensions, see CNC Policy Metrics.

9.15 LDAP Gateway

The following table describes the LDAP Gateway Service metrics and respective dimensions:

Table 9-251 ldap_request_total

Field	Details
Description	Total number of LDAP requests.
Type	Counter
Dimension	• code • ReqType

Example: ldap_request_total{ReqType="GET",application="ldapgateway"} 13.0 ; Type-Counter

Table 9-252 ldap_response_total

Field	Details
Description	Total number of LDAP responses.
Type	Counter
Dimension	• ReqType • Code • application

Examples:

- ldap_response_total{Code="4xx",ReqType="GET",application="ldapgateway"} 0.0 ; Type-Counter
- ldap_response_total{Code="2xx",ReqType="GET",application="ldapgateway"} 13.0 ; Type-Counter
- ldap_response_total{Code="5xx",ReqType="GET",application="ldapgateway"} 0.0 ; Type-Counter

Table 9-253 topic_version

Field	Details
Description	Describes the current applied version of a given topic (mentioned in dimension topic_name) into the pod.
Type	Gauge
Dimension	• Service Name • Pod Name

For more information about Dimensions, see CNC Policy Metrics.

9.16 Binding Service Metrics

The following table describes the Binding Service metrics and respective dimensions:

Table 9-254 ocpm_binding_inbound_request_total

Field	Details
Description	Total number of inbound requests.
Type	Counter
Dimension	• operation_type • servicename_3gpp
Examples	• ocpm_binding_inbound_request_total{operation_type="read",nfInstanceId="3db4b97c-04dc-4aff-ab20-2e156dbd02d2",nf_name="bsf.oracle.com",servicename_3gpp="binding.1.0",} 5.0 ; Type-Counter • ocpm_binding_inbound_request_total{operation_type="delete",nfInstanceId="3db4b97c-04dc-4aff-ab20-2e156dbd02d2",nf_name="bsf.oracle.com",servicename_3gpp="binding.1.0",} 10.0 ; Type-Counter • ocpm_binding_inbound_request_total{operation_type="write",nfInstanceId="3db4b97c-04dc-4aff-ab20-2e156dbd02d2",nf_name="bsf.oracle.com",servicename_3gpp="binding.1.0",} 91.0 ; Type-Counter

Table 9-255 ocpm_egress_request_total

Field	Details
Description	Total number of egress requests.
Type	Counter
Dimension	• operation_type • servicename_3gpp • nf_instance_id • servicename_3gpp • nf_name • egress_service

Table 9-256 ocpm_egress_response_total

Field	Details
Description	Total number of egress responses.
Type	Counter
Dimension	• servicename_3gpp • response_code • nf_instance_id • nf_name • egress_service • latency

Table 9-257 ocpm_binding_inbound_response_total

Field	Details
Description	Total number of inbound responses.
Type	Counter

Table 9-257 (Cont.) ocpm_binding_inbound_response_total

Field	Details
Dimension	• operation • task • response_code
Examples	• ocpm_binding_inbound_response_total{operation_type="delete",nfInstanceId="3db4b97c-04dc-4aff-ab20-2e156dbd02d2",nf_name="bsf.oracle.com",response_code="2xx",servicename_3gpp="binding.1.0",} 10.0 ; Type-Counter • ocpm_binding_inbound_response_total{operation_type="write",nfInstanceId="3db4b97c-04dc-4aff-ab20-2e156dbd02d2",nf_name="bsf.oracle.com"response_code="2xx",servicename_3gpp="binding.1.0",} 91.0 ; Type-Counter • ocpm_binding_inbound_response_total{operation_type="read",nfInstanceId="3db4b97c-04dc-4aff-ab20-2e156dbd02d2",nf_name="bsf.oracle.com"response_code="4xx",servicename_3gpp="binding.1.0",} 1.0 ; Type-Counter • ocpm_binding_inbound_response_total{operation_type="read",nfInstanceId="3db4b97c-04dc-4aff-ab20-2e156dbd02d2",nf_name="bsf.oracle.com"response_code="5xx",servicename_3gpp="binding.1.0",} 1.0 ; Type-Counter

Table 9-258 occnp_binding_query_request_count

Field	Details
Description	This metric tracks the total number of Audit Notify requests received from BSF.
Type	Counter
Dimension	• operation • servicename_3gpp
Example	occnp_binding_query_request_count_total{ingress_service="",operation_type="AuditNotify",servicename_3gpp="binding",} 1.0

Table 9-259 occnp_binding_query_response_count

Field	Details
Description	This metric tracks the total number of successful responses sent to BSF for Audit Notify requests.
Type	Counter
Dimension	• operation • servicename_3gpp • response_code
Example	occnp_binding_query_response_count_total{ingress_service="",operation_type="AuditNotify",response_code="2xx",servicename_3gpp="binding",} 1.0

Table 9-260 occnp_audit_notif_request_count

Field	Details
Description	The total number of audit notification requests received by binding service.
Type	Counter

Table 9-260 (Cont.) occnp_audit_notif_request_count

Field	Details
Dimension	• servicename_3gpp • nfInstanceId
Example	occnp_audit_notif_request_count_total{nfInstanceId="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",servicename_3gpp="binding",} 1.0 ; Type-Counter

Table 9-261 occnp_audit_notif_response_count

Field	Details
Description	The total number of responses sent by binding service for audit notifications.
Type	Counter
Dimension	• servicename_3gpp • nfInstanceId • response_code
Example	occnp_audit_notif_response_count_total{nfInstanceId="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",response_code="2xx",servicename_3gpp="binding",} 1.0 ; Type-Counter

Table 9-262 occnp_stale_session_cleanup_count

Field	Details
Description	Total number of records deleted from the binding database by the stale session clean up process.
Type	Counter
Dimension	• servicename_3gpp • sessionType
Example	occnp_stale_session_cleanup_count_total{servicename_3gpp="binding",sessionType="dependentcontextbinding",} 1.0 ; Type-Counter

Table 9-263 topic_version

Field	Details
Description	Describes the current applied version of a given topic (mentioned in dimension topic_name) into the pod.
Type	Gauge
Dimension	• Service Name • Pod Name

Table 9-264 occnp_binding_service_overall_processing_time_seconds

Field	Details
Description	Binding service overall processing time.
Type	Summary

Table 9-264 (Cont.) occnp_binding_service_overall_processing_time_seconds

Field	Details
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Table 9-265 occnp_binding_service_overall_processing_time_seconds_max

Field	Details
Description	Binding service overall processing time.
Type	Gauge
Dimension	• application • error • exception • hostname • method • outcome • status • uri
Examples	• occnp_binding_service_overall_processing_time_seconds_max{error="none", exception="none", method="POST", outcome="SUCCESS", status="201", uri="/binding/v1/contextBinding/context-owner/{contextOwner}",} 2.595594434 • occnp_binding_service_overall_processing_time_seconds_max{error="none", exception="none", method="DELETE", outcome="SUCCESS", status="204", uri="/binding/v1/contextBinding/contextId/{contextId}",} 0.276239293 • occnp_binding_service_overall_processing_time_seconds_max{error="ContextBindingNotFound", exception="ContextBindingNotFound", method="DELETE", outcome="CLIENT_ERROR", status="404", uri="/binding/v1/contextBinding/cleanup",} 0.115288423

Table 9-266 occnp_binding_service_overall_processing_time_seconds_count

Field	Details
Description	Binding service overall processing time.
Type	Counter
Dimension	• application • error • exception • hostname • method • outcome • status • uri

Table 9-266 (Cont.) occnp_binding_service_overall_processing_time_seconds_count

Field	Details
Examples	• occnp_binding_service_overall_processing_time_seconds_count{error="none", exception="none", method="POST", outcome="SUCCESS", status="201", uri="/binding/v1/contextBinding/context-owner/{contextOwner}",} 3.0 • occnp_binding_service_overall_processing_time_seconds_count{error="none", exception="none", method="DELETE", outcome="SUCCESS", status="204", uri="/binding/v1/contextBinding/contextId/{contextId}",} 3.0 • occnp_binding_service_overall_processing_time_seconds_count{error="ContextBindingNotFound", exception="ContextBindingNotFound",method="DELETE",outcome="CLIENT_ERROR", status="404",uri="/binding/v1/contextBinding/cleanup",} 6.0

Table 9-267 occnp_binding_service_overall_processing_time_seconds_sum

Field	Details
Description	Binding service overall processing time.
Type	Counter
Dimension	• application • error • exception • hostname • method • outcome • status • uri
Examples	• occnp_binding_service_overall_processing_time_seconds_sum{error="none",exception="none",method="POST", outcome="SUCCESS", status="201", uri="/binding/v1/contextBinding/context-owner/{contextOwner}",} 5.861114721 • occnp_binding_service_overall_processing_time_seconds_sum{error="none", exception="none",method="DELETE",outcome="SUCCESS",status="204",uri="/binding/v1/contextBinding/contextId/{contextId}",} 0.349598291 • occnp_binding_service_overall_processing_time_seconds_sum{error="ContextBindingNotFound",exception="ContextBindingNotFound", method="DELETE",outcome="CLIENT_ERROR",status="404", uri="/binding/v1/contextBinding/cleanup",} 0.203103476

Table 9-268 occnp_bindingservice_db_operation_time_seconds_count

Field	Details
Description	Binding service overall database processing time.
Type	Gauge
Dimension	• exception • method • repository • state
Examples	• occnp_bindingservice_db_operation_time_seconds_count{exception="None",method="findByTopicInfo",repository="ConfigurationItemRepository",state="SUCCESS",} 7721880.0 • occnp_bindingservice_db_operation_time_seconds_count{exception="None",method="findByName",repository="TopicInfoRepository",state="SUCCESS",} 8190322.0

Table 9-269 occnp_bindingservice_db_operation_time_seconds_sum

Field	Details
Description	Binding service overall database processing time.
Type	Gauge
Dimension	• exception • method • repository • state
Examples	• occnp_bindingservice_db_operation_time_seconds_sum{exception="None",method="findByTopicInfo",repository="ConfigurationItemRepository",state="SUCCESS",} 4713.01 • occnp_bindingservice_db_operation_time_seconds_sum{exception="None",method="findByName",repository="TopicInfoRepository",state="SUCCESS",} 6328.74490

Table 9-270 occnp_bindingservice_db_operation_time_seconds_max

Field	Details
Description	Binding service overall database processing time.
Type	Gauge
Dimension	• exception • method • repository • state

Table 9-271 ocpm_egress_request_timeout_total

Field	Details
Description	Binding service overall database processing time.
Type	Gauge
Dimension	• exception • method • repository • state
Examples	• ocpm_egress_request_timeout_total{Host_Name="",application="pcf_bindingservice",egress_service="BSF",nfInstanceId="3db4b97c-04dc-4aff-ab20-2e156dbd02d2",nf_name="bsf.oracle.com",operation_type="write",service_resource="pcfBindings",service_version="V1",servicename_3gpp="binding",} 1.0 • ocpm_egress_request_timeout_total{Host_Name="",application="pcf_bindingservice",egress_service="BSF",nfInstanceId="3db4b97c-04dc-4aff-ab20-2e156dbd02d2",nf_name="bsf.oracle.com",operation_type="delete",service_resource="pcfBindings",service_version="V1",servicename_3gpp="binding",} 1.0

Table 9-272 occnp_stale_session_cleanup_internal_queue_size

Field	Details
Description	Bindiing session stale internal queue size.
Type	Gauge
Dimension	servicename_3gpp

Table 9-272 (Cont.) occnp_stale_session_cleanup_internal_queue_size

Field	Details
Example	occnp_stale_session_cleanup_internal_queue_size{servicename_3gpp="binding",} 2.0; Type-Gauge

Table 9-273 occnp_audit_delete_record_count

Field	Details
Description	This metric is used to monitor records deleted in binding by Audit Service.
Type	Counter
Dimension	• max_ttl_reached (true/false) • nfInstanceId • sessionType (contextbinding/ dependentContextBinding) • servicename_3gpp
Examples	occnp_audit_delete_record_count{max_ttl_reached="true",nfInstanceId="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",servicename_3gpp="binding",sessionType="contextbinding",}2.0

Table 9-274 error_handler_in_total

Field	Details
Description	This metric is incremented on initialization of error handling.
Type	Counter
Dimensions	• application • applicationException • wrapped_exception
Example	occnp_error_handler_in_total{application_exception="JavaException",status="500",wrapped_exception="ServiceException",} 1.0

Table 9-275 error_handler_exec_total

Field	Details
Description	This metric is incremented on error handling performed by framework.
Type	Counter
Dimensions	• application • applicationException • error_type • operation • origin • rule_name • source_interface • target_interface • wrapped_exception
Example	occnp_error_handler_exec_total{application_exception="JavaException",error_type="INTERNAL",operation="AUDIT",origin="JAVA",rule_name="REJECT_WITH_ENHANCED_DETAIL",source_interface="BSF",status="404",target_interface="POLICY",wrapped_exception="ServiceException",} 1.0

Table 9-276 error_handler_out_total

Field	Details
Description	This metric is incremented on completion of error handling.
Type	Counter
Dimensions	• application • applicationException • error_resolved • wrapped_exception
Example	occnp_error_handler_out_total{application_exception="JavaException",error_resolved="true",status="404",wrapped_exception="ServiceException",} 1.0

For more information about Dimensions, see CNC Policy Metrics.

9.17 Audit Service Metrics

The following table describes the Audit Service metrics and respective dimensions:

Table 9-277 audit_recs_stale_total

Field	Details
Description	Number of records detected as stale.
Type	Counter
Dimension	• ServiceName • TableName
Example	audit_recs_stale_total{ServiceName="pcf-ueservice",TableName="UePolicyAssociation",} 1.0

Table 9-278 audit_recs_notif_total

Field	Details
Description	Number of stale record notifications sent, applicable for modes: NOTIFY and DELETE_NOTIFY
Type	Counter
Dimension	ServiceName
Examples	• audit_recs_notif_total{ServiceName="sm-service"} 50.0 ; Type-Counter • audit_recs_notif_total{ServiceName="pcf-ueservice",} 1.0

Table 9-279 audit_recs_remv_total

Field	Details
Description	Number of stale records deleted, applicable for modes: DELETE and DELETE_NOTIFY.
Type	Counter
Dimension	• ServiceName • TableName
Example	audit_recs_remv_total{ServiceName="sm-service",TableName="SmPolicyAssociation"} 5.0 ; Type-Counter

Table 9-280 audit_recs_remv_ex_total

Field	Details
Description	Number of exceptions hit during attempt to delete a stale record.
Type	Counter
Dimension	• ServiceName • TableName
Example	audit_recs_remv_ex_total{ServiceName="sm-service",TableName="SmPolicyAssociation"} 0.0 ; Type-Counter

Table 9-281 audit_recs_notif_ex_total

Field	Details
Description	Number of exceptions hit during attempt to notify.
Type	Counter
Dimension	ServiceName
Example	audit_recs_notif_ex_total{ServiceName="sm-service"} 0.0 ; Type-Counter

Table 9-282 audit_recs_notif_err_total

Field	Details
Description	Number of error responses received for notifications sent.
Type	Counter
Dimension	ServiceName
Example	audit_recs_notif_err_total{ServiceName="sm-service"} 13.0 ; Type-Counter

Table 9-283 audit_recs_deque_for_notif_total

Field	Details
Description	Number of stale records dequeued to send notification.
Type	Counter
Dimension	ServiceName
Example	audit_recs_deque_for_notif_total{ServiceName="pcf-ueservice",} 1.0

Table 9-284 audit_recs_enque_for_notif_total

Field	Details
Description	Number of stale records enqueued from database.
Type	Counter
Dimension	ServiceName
Example	audit_recs_enque_for_notif_total{ServiceName="pcf-ueservice",} 1.0

Table 9-285 audit_recs_enque_err_total

Field	Details
Description	Number of stale records failed to enqueue if Queue is full.
Type	Counter
Dimension	ServiceName
Example	audit_recs_enque_err_total{ServiceName="sm-service"} 0.0 ; Type-Counter

Table 9-286 oc_db_active_session_count

Field	Details
Description	Reports the session for a given service e.g. SM session.
Type	Gauge
Dimension	• ServiceName • TableName
Examples	• oc_db_active_session_count{Service="sm-service",Table="smassociation",} 177.0 ; Type-Gauge • oc_db_active_session_count{Service="sm-service",Table="Rx",} 0.0 ; Type-Gauge • oc_db_active_session_count{Service="am-service",Table="amassociation",} 0.0 • oc_db_active_session_count{Service="pcrf-core",Table="gxsession",} 0.0 ; • oc_db_active_session_count{Service="pcrf-core",Table="rxsession",} 0.0 ; • oc_db_active_session_count{Service="pcf-ueservice",TableName="UePolicyAssociation",} 1.0

Table 9-287 oc_db_records_count

Field	Details
Description	Reports the number of records in table that is being monitored by audit service.
Type	Gauge
Dimension	• ServiceName • TableName
Example	oc_db_records_count{Service="sm-service",Table="SiteJsonSchemaVersionInfo",} 4.0 ; Type-Gauge

Table 9-288 occnp_pending_binding_reattempts_total

Field	Details
Description	Indicates when a attempt to recreate has been triggered by PCF.
Type	Counter
Dimension	Attempt

Table 9-289 occnp_pending_binding_terminate_all_attempts_failed_total

Field	Details
Description	Indicates when session terminate has been triggered when all recreate attempts fail.
Type	Counter

Table 9-289 (Cont.) occnp_pending_binding_terminate_all_attempts_failed_total

Field	Details
Dimension	• RESPONSE_CODE, • CAUSE

Table 9-290 occnp_pending_binding_reattempt_fail_total

Field	Details
Description	Indicates when each recreate attempt fail.
Type	Counter
Dimension	NA

Table 9-291 occnp_pending_operation_threshold_reached_total

Field	Details
Description	Indicates when pending operation table threshold has been reached.
Type	Counter
Dimension	NA

Table 9-292 occnp_reject_sm_create_threshold_reached_total

Field	Details
Description	Indicates when SM rejects SM CREATE when pending operation threshold has been reached.
Type	Counter
Dimension	• DNN • CAUSE • RESPONSE_CODE

Table 9-293 occnp_pending_operation_records_count

Field	Details
Description	Indicates the amount of entries in the PendingOperation table.
Type	Gauge
Dimension	NA

Table 9-294 topic_version

Field	Details
Description	Describes the current applied version of a given topic (mentioned in dimension topic_name) into the pod.
Type	Gauge
Dimension	• Service Name • Pod Name

For more information about Dimensions, see CNC Policy Metrics.

9.18 Query Service Metrics

The following table describes the Query Service metrics and respective dimensions:

Table 9-295 queryservice_sessionDelete_request_total

Field	Details
Description	Total number of delete SM service query requests.
Type	Counter
Dimension	• DeleteScope • ResourceLevel • ResourceType
Examples	• queryservice_sessionDelete_request_total{DeleteScope="Local",ResourceLevel="Individual",ResourceType="BINDING"} 1.0 ; Type-Counter • queryservice_sessionDelete_request{"name":"queryservice_sessionDelete_request","measurements":[{"statistic":"COUNT","value":2.0}],"availableTags":[{"tag":"DeleteScope","values":["Local"]},{"tag":"ResourceLevel","values":["Individual"]},{"tag":"ResourceType","values":["SM"]}]}

Table 9-296 queryservice_sessionDelete_response_total

Field	Details
Description	Total number of responses for delete SM service query.
Type	Counter
Dimension	• DeleteScope • ResourceLevel • ResourceType • Result_Code
Examples	• queryservice_sessionDelete_response_total{DeleteScope="Local",ResourceLevel="Individual",ResourceType="BINDING",Result_Code="2XX"} 1.0 ; Type-Counter • queryservice_sessionDelete_response{"name":"queryservice_sessionDelete_response","measurements":[{"statistic":"COUNT","value":2.0}],"availableTags":[{"tag":"DeleteScope","values":["Local"]},{"tag":"ResourceLevel","values":["Individual"]},{"tag":"ResourceType","values":["SM"]},{"tag":"Result_Code","values":["2XX"]}]}

For more information about Dimensions, see CNC Policy Metrics.

9.19 AppInfo Metrics

This section describes the metrics and examples for AppInfo service.

Table 9-297 appinfo_service_running

Field	Details
Description	Provides the status of monitored services.
Type	Gauge

Table 9-297 (Cont.) appinfo_service_running

Field	Details
Dimension	• service • namespace • category
Example	appinfo_service_running{service="xxx",namespace="xxx",category="xxx"} 1

Table 9-298 appinfo_category_running

Field	Details
Description	Provides the status of monitored categories
Type	Gauge
Dimension	• namespace • category
Example	appinfo_category_running{category="xxx",namespace="xxx"} 1

Table 9-299 appinfo_category_good

Field	Details
Description	Provides the readiness of monitored categories.
Type	Gauge
Dimension	• namespace • category
Example	appinfo_category_good{category="xxx",namespace="xxx"} 1

Table 9-300 nfscore

Field	Details
Description	*factor*: • Contains one of the values: all, successTPS, serviceHealth, signallingConnections, replicationHealth, localityPreference. • When the factor is set to *all* that means NF score is calculated for all the factors. *status*: • *success*: when the factor is enabled and its value is fetched successfully. • *failed*: when the factor is enabled and fetching the value fails. • *notCalculated*: when the factor is disabled.
Type	Gauge
Dimension	• nfInstanceID • factor • Status

Table 9-300 (Cont.) nfscore

Field	Details
Example	nfscore{app="testing-appinfo", app_kubernetes_io_instance="testing", app_kubernetes_io_managed_by="Helm", app_kubernetes_io_name="appinfo", app_kubernetes_io_part_of="ocbsf", app_kubernetes_io_version="1.7.1.0.0", application="ocbsf", calculatedStatus="success", container="appinfo", endpoint="cnc-metrics", engVersion="22.4.0", factor="localityPreference", helm_sh_chart="appinfo-22.4.0", instance="10.233.117.146:9000", job="occne-infra/occne-nf-cnc-podmonitor", microservice="bsf-app-info", mktgVersion="1.7.1.0.0", namespace="biloxi-ns", nfInstanceID="fe7d992b-0541-4c7d-ab84-c6d70b1b0666", pod="testing-appinfo-78dc65865f-hgrhk", pod_template_hash="78dc65865f", vendor="Oracle"} 5

Table 9-301 nfScoringFactorActualValue

Field	Details
Description	The factor tag would contain one of the following values: successTPS, serviceHealth, signallingConnections, replicationHealth, localityPreference
Type	Gauge
Dimension	• nfInstanceID • factor
Example	nfScoringFactorActualValue{app="testing-appinfo", app_kubernetes_io_instance="testing", app_kubernetes_io_managed_by="Helm", app_kubernetes_io_name="appinfo", app_kubernetes_io_part_of="ocbsf", app_kubernetes_io_version="1.7.1.0.0", application="ocbsf", calculatedStatus="success", container="appinfo", endpoint="cnc-metrics", engVersion="22.4.0", factor="localityPreference", helm_sh_chart="appinfo-22.4.0", instance="10.233.117.146:9000", job="occne-infra/occne-nf-cnc-podmonitor", microservice="bsf-app-info", mktgVersion="1.7.1.0.0", namespace="biloxi-ns", nfInstanceID="fe7d992b-0541-4c7d-ab84-c6d70b1b0666", pod="testing-appinfo-78dc65865f-hgrhk", pod_template_hash="78dc65865f", vendor="Oracle"} 5

> **Note**
>
> Sh interface is not supported for Converged Policy mode of deployment.

For more information about dimensions, see CNC Policy Metrics.

9.20 PerfInfo Metrics

This section describes the metrics and examples for PerfInfo service.

Table 9-302 nf_load_info

Field	Details
Description	Provides information about service load.
Type	Gauge
Dimension	• service • namespace
Example	nf_load_info{namespace="xxx",service="xxx"} 0.8486912141984638

Table 9-303 jvm_cpu_usage

Field	Details
Description	Springboot per service jvm_cpu_usage.
Type	Gauge
Dimension	• service • namespace
Example	jvm_cpu_usage{namespace="xxx",service="xxx"} 0.2758240242725142

Table 9-304 jvm_memory

Field	Details
Description	Springboot per service jvm_memory
Type	Gauge
Dimension	• service • namespace
Example	jvm_memory{namespace="ttz",service="xxx"} 18.361382484436035

Table 9-305 cgroup_cpu_nanoseconds

Field	Details
Description	Reports the total CPU time (in nanoseconds) on each CPU core for all the tasks in the cgroup.
Type	Gauge
Dimension	NA
Example	cgroup_cpu_nanoseconds 2.1782821080274e+013

Table 9-306 cgroup_memory_bytes

Field	Details
Description	Reports the memory usage.
Type	Gauge
Dimension	NA
Example	cgroup_memory_bytes 1.31289088e+08

Table 9-307 load_level

Field	Details
Description	This metric provides information about the load level of a service.
Type	Gauge
Dimension	• service • namespace • isLeaderPod
Example	load_level{serivce="xxx"} L1

Table 9-308 service_resource_stress

Field	Details
Description	This metric tracks CPU, memory, failure count, and pending requests on the basis of which the overload level of a service is calculated.
Type	Gauge
Dimension	• type • service • namespace • isLeaderPod
Example	service_resource_stress{service="xxx", type="xxx"}10.0

Table 9-309 service_resource_overload_level

Field	Details
Description	This metric tracks an individual resource's overload level that is calculated based on the resource usage and configured threshold.
Type	Gauge
Dimension	• type • service • namespace • isLeaderPod
Example	service_resource_overload_level{service="xxx", type="xxx"}2.0

Table 9-310 system_overload_threshold_config_mode

Field	Details
Description	Indicates whether the overload level threshold configuration is based on STANDALONE or PROFILE mode.
Type	Gauge
Dimension	• namespace • isLeaderPod
Example	system_overload_threshold_config_mode 1.0

Table 9-311 active_overload_threshold_fetch_failed

Field	Details
Description	Indicates whether the active profile data is fetched successfully or failed to fetch.
Type	Gauge
Dimension	• namespace • isLeaderPod The value of this dimension can be either 0 or 1. Where 0, represents "Successfully fetched the active threshold"and 1 represents "Failure in fetching the active threshold".
Example	active_overload_threshold_fetch_failed 1.0

Table 9-312 oc_ingressgateway_route_overloadcontrol_discard

Field	Details
Description	This metric is pegged when incoming request is discarded by OverloadLoad Filter on priority or percentage basis.
Type	Counter
Dimension	• Method • ServiceName • Status • discard policy: priority basis OR percentage basis • Scheme • InstanceIdentifier • ErrorOriginator • namespace
Example	

Table 9-313 load_level_report_total

Field	Details
Description	This metric is used to track: • the number of times load level calculation is performed • the number of times load level changes • how long the particular level was active
Type	Counter
Dimension	• level • service • prevLevel • namespace • levelChangeType • isLeaderPod
Example	load_level_report_total{namespace="hi-riley", service="pcf-occnp-pcrf-core", isLeaderPod="True", level="Normal", levelChangeType="-"} 2.0

Table 9-314 service_resource_overload_level_report_total

Field	Details
Description	This metric is used to track: • the number of times load level calculation is performed. • the number of times load level changes. • for how long the particular level was active for each metric type.
Type	Counter

Table 9-314 (Cont.) service_resource_overload_level_report_total

Field	Details
Dimension	• level Possible values: – L1 – L2 – L3 – Normal • service • prevLevel • levelChangeType Possible values: – increment – decrement • isLeaderPod Possible values: – True – False • type Possible values: – cpu – memory – svc_failure_count – svc_pending_count • namespace
Example	service_resource_overload_level_report_total{namespace="hi-riley", service="pcf-occnp-pcrf-core", isLeaderPod="True", level="Normal", levelChangeType="-", type="cpu"} 2.0

Table 9-315 http_out_conn_request

Field	Details
Description	This counter metric is used to count the number of http API Egress requests.
Type	Counter
Dimension	• servicenameNon3gpp • serviceResource • serviceVersion • operationType • namespace • params • isLeaderPod
Example	

Table 9-316 http_out_conn_response

Field	Details
Description	This counter metric is used to count the number of http API Egress responses.
Type	Counter

Table 9-316 (Cont.) http_out_conn_response

Field	Details
Dimension	• servicenameNon3gpp • serviceResource • serviceVersion • outcome • namespace • operationType • responseCode • params • cause • isLeaderPod
Example	

Table 9-317 overload_manager_enabled

Field	Details
Description	This metric indicates whether overload manager is enabled or disabled.
Type	Gauge
Dimension	• source Possible values: – DIAM_GW – INGRESS_GW – PERF_INFO • namespace
Example	overload_manager_enabled{namespace="hi-riley", source="DIAM_GW"} 1

Table 9-318 leader_pod

Field	Details
Description	This metric is used to know the leader pod.
Type	Gauge
Dimension	• namespace
Example	leader_pod{namespace="hi-riley"} 1

For more information about dimensions, see CNC Policy Metrics.

9.21 Pod Congestion Metrics

This section describes the metrics and examples for pod congestion control.

Note

- **Current behavior in Policy services supporting Pod Congestion Observability Gauge Metrics:**

  In Grafana dashboard, the congestion state metric (occnp_pod_congestion_state, occnp_pod_resource_congestion_state) and rejection state metric (um_http_congestion_message_reject_total) graphs show inconsistent behavior. As in the pod congestion state graph shows to be in normal state while discard graph does not show flat line plot but with discard messages.

  This is caused because pods with the CPU usage level on the borderline of DANGER_OF_CONGESTION (DOC) level with sudden CPU spikes, pushes the pod to DANGER_OF_CONGESTION (DOC) state or CONGESTION_L1 state. During this CPU spike, based on the Load Shedding rule configured in CNC Console the specific requests are discarded and rejection counter metric is incremented. After the brief spike interval the congestion state is reported in Grafana dashboard as normal.

  Prometheus scrape metrics from its targeted instance periodically based on scrape interval. But the scrape interval is not granular enough to capture the abrupt CPU spikes. um_http_congestion_message_reject_total is visualized by using rate() function over a time duration. Due to the above cause the counter metrics increments and when averaged over a time interval the rejection plot seems to be constant although it was incremented for the requests in a few seconds when pod was in not normal state due to abrupt CPU spikes.

- **Prefix and Suffix support for `pod_cong_state_report` and `pod_resource_congestion_state_report` metrics**

  In Policy 24.2.x, the `pod_cong_state_report` and `pod_resource_congestion_state_report` metrics supports prefix and suffix. If users have created dashboard in 24.1.x using these metrics, then they should update their dashboards by adding prefix and suffix for these metrics. The default value prefix is `occnp` and suffix is empty string.

Table 9-319 occnp_pod_congestion_state

Field	Details
Description	Tracks congestion state of pod
Type	Gauge
Dimension	level = 0,1,2 (0 = Normal, 1 = DoC, 2 = Congested) are for Bulwark and Diameter Gateway services. level = 0,1,2,3,4 (0=Normal,1=DoC,2=Congestion_L1,3=Congestion_L2,4=Congested) are for other Policy services.
Examples	• occnp_pod_congestion_state 0.0 • occnp_pod_congestion_state 1.0

Table 9-320 occnp_pod_resource_stress

Field	Details
Description	Tracks CPU, memory, queue usage (pending requests) for binding, pds, usage monitoring and SM serivces.

Table 9-320 (Cont.) occnp_pod_resource_stress

Field	Details
Type	Gauge
Dimension	type = "Queue","CPU","Memory"
Examples	• occnp_pod_resource_stress{type="memory",} 35.0 • occnp_pod_resource_stress{type="cpu",} 50.0

Table 9-321 occnp_pod_resource_congestion_state

Field	Details
Description	Tracks individual resource's congestion state calculated based on the resource usage and configured threshold
Type	Gauge
Dimension	• type = "Queue","CPU","Memory" • level = 0,1,2 (0 = Normal, 1 = DoC, 2 = Congested) for Bulwark and Diameter Gateway services level = 0,1,2,3,4 (0=Normal,1=DoC,2=Congestion_L1,3=Congestion_L2,4=Congested) are for other Policy services
Examples	• occnp_pod_resource_congestion_state{type="memory",} 0.0 • occnp_pod_resource_congestion_state{type="cpu",} 1.0

Table 9-322 diam_congestion_message_reject_total

Field	Details
Description	Tracks number of messages rejected due to congestion
Type	Counter
Dimension	• priority = calculated or received DRMP priority of message being rejected • response_code = response code sent with rejected message • destHost • destRealm • origHost • origRealm • appId • cmdCode • msgType • direction
Example	diam_congetion_msg_reject_total{appId="16777236",cmdCode="265",destHost="",destRealm="example.com",msgType="request",origHost="diamcliaf.example.com",origRealm="test.example.com",priority="12",responseCode="3004",} 1.0

Table 9-323 pod_cong_state_report_total

Field	Details
Description	Track the total count of change of congestion state.
Type	Counter

Table 9-323 (Cont.) pod_cong_state_report_total

Field	Details
Dimension	• namespace • pod • old_state • new_state
Example	--

Table 9-324 pod_resource_congestion_state_report_total

Field	Details
Description	Track the total count of change of resource specific congestion state.
Type	Counter
Dimension	• resourceType • namespace • pod • old_state • new_state
Example	--

Table 9-325 http_congestion_message_reject_total

Field	Details
Description	Track the rejected messages due to pod congestion.
Type	Counter
Dimension	• priority • responseCode • operation • uri
Example	--

Table 9-326 um_http_congestion_message_reject_total

Field	Details
Description	Usage Monitoring service tracks the rejected messages due to pod congestion.
Type	Counter
Dimension	• priority • responseCode • operation • uri
Examples	--

For more information about dimensions, see CNC Policy Metrics.

9.22 PCRF Core Metrics

This section describes the metrics and examples for PCRF Core Metrics.

General Metrics

Table 9-327 occnp_pcrf_core_overall_processing_time_sum_total

Field	Details
Description	Sum of pcrf core overall processing time.
Type	Counter
Dimension	• method • status
Example	occnp_pcrf_core_overall_processing_time_sum_total{method="auditNotification",status="200 OK",} 299.0

Table 9-328 occnp_pcrf_core_overall_processing_time_count_total

Field	Details
Description	Total number of pcrf core overall processing time.
Type	Counter
Dimension	• method • status
Example	occnp_pcrf_core_overall_processing_time_count_total{method="auditNotification",status="200 OK",} 16.0

Table 9-329 http_server_requests_seconds_count

Field	Details
Description	Number of http request received.
Type	Counter
Dimension	• method • outcome • status • uri
Example	http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/pcrfSessionData",} 2.0

Table 9-330 http_server_requests_seconds_max

Field	Details
Description	Maximum number of http requests received.
Type	Counter
Dimension	• method • outcome • status • uri

Table 9-330 (Cont.) http_server_requests_seconds_max

Field	Details
Example	http_server_requests_seconds_max{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/pcrfSessionData",} 0.0

Table 9-331 topic_version

Field	Details
Description	Describes the current applied version of a given topic (mentioned in dimension topic_name) into the pod.
Type	Gauge
Dimension	• Service Name • Pod Name

Diameter Metrics

Table 9-332 occnp_diam_request_local_total (CCR-I)

Field	Details
Description	Number of , CCA-I initial request sent.
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType = (CCR-I, CCA-I • origHost • origRealm
Example	occnp_diam_request_local_total{appId="16777238",appType="Gx",cmdCode="272",destHost="",destRealm="oracle.com",direction="in",msgType="CCR-I",origHost="pgw1.oracle.com",origRealm="oracle.com",} 1.0 1633975504473

Table 9-333 occnp_app_request_local_process_total

Field	Details
Description	Number of requests sent to PGW during RAR flow along with number of retries sent to PGW.
Type	Counter
Dimension	• appType • destHost • destRealm • msgType • reason • retryAttempts
Example	occnp_app_request_local_process_total{appType="Gx",destHost="pgw.oracle.com",destRealm="oracle.com",msgType="RAR",reason="notification",retryAttempts="1",} 1.0

Table 9-334 occnp_app_response_local_process_total

Field	Details
Description	Number of response received from PGW during RAR flow with response code and cause.
Type	Counter
Dimension	• appType • errorType • msgType • reason • reqDestHost • reqDestRealm • responseType • retryAttempts
Example	occnp_app_response_local_process_total{appType="Gx",errorType="",msgType="RAA",reason="notification",reqDestHost="pgw.oracle.com",reqDestRealm="oracle.com",responseType="DIAMETER_SUCCESS",retryAttempts="1",} 1.0

Table 9-335 occnp_diam_response_local_total (CCA-I)

Field	Details
Description	Number of CCA initial response received.
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm • reqDestHost • reqDestRealm • reqOrigHost • reqOrigRealm • responseCode
Example	occnp_diam_response_local_total{appId="16777238",appType="Gx",cmdCode="272",destHost="",destRealm="",direction="out",msgType="CCA-I",origHost="XYZ-IN.us.oracle.com",origRealm="oracle.com",reqDestHost="",reqDestRealm="oracle.com",reqOrigHost="pgw1.oracle.com",reqOrigRealm="oracle.com",responseCode="2001",} 1.0 1633975504473

Table 9-336 occnp_diam_request_local_total (CCR-U)

Field	Details
Description	Number of CCR update request sent.
Type	Counter

Table 9-336 (Cont.) occnp_diam_request_local_total (CCR-U)

Field	Details
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm
Example	occnp_diam_request_local_total{appId="16777238",appType="Gx",cmdCode="272",destHost="XYZ-IN.us.oracle.com",destRealm="oracle.com",direction="in",msgType="CCR-U",origHost="pgw1.oracle.com",origRealm="oracle.com",} 1.0 1633975504473

Table 9-337 occnp_diam_response_local_total (CCA-U)

Field	Details
Description	Number of CCA update response received.
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm • reqDestHost • reqDestRealm • reqOrigHost • reqOrigRealm • responseCode
Example	occnp_diam_response_local_total{appId="16777238",appType="Gx",cmdCode="272",destHost="",destRealm="",direction="out",msgType="CCA-U",origHost="XYZ-IN.us.oracle.com",origRealm="oracle.com",reqDestHost="XYZ-IN.us.oracle.com",reqDestRealm="oracle.com",reqOrigHost="pgw1.oracle.com",reqOrigRealm="oracle.com",responseCode="2001",} 1.0 1633975504473

Table 9-338 occnp_diam_request_local_total (CCR-T)

Field	Details
Description	Number of CCR terminate request sent.
Type	Counter

Table 9-338 (Cont.) occnp_diam_request_local_total (CCR-T)

Field	Details
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm
Example	• occnp_diam_request_local_total{appId="16777238",appType="Gx",cmdCode="272",destHost="",destRealm="oracle.com",direction="in",msgType="CCR-T",origHost="pgw1.oracle.com",origRealm="oracle.com",} 1.0 1633975504473 • occnp_diam_request_local_total{appId="16777238",appType="Gx",cmdCode="272",destHost="XYZ-IN.us.oracle.com",destRealm="oracle.com",direction="in",msgType="CCR-T",origHost="pgw1.oracle.com",origRealm="oracle.com",} 1.0 1633975504473

Table 9-339 occnp_diam_response_local_total (CCA-T)

Field	Details
Description	Number of CCA terminate response received.
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm • reqDestHost • reqDestRealm • reqOrigHost • reqOrigRealm • responseCode
Example	• occnp_diam_response_local_total{appId="16777238",appType="Gx",cmdCode="272",destHost="",destRealm="",direction="out",msgType="CCA-T",origHost="XYZ-IN.us.oracle.com",origRealm="oracle.com",reqDestHost="",reqDestRealm="oracle.com",reqOrigHost="pgw1.oracle.com",reqOrigRealm="oracle.com",responseCode="5002",} 1.0 1633975504473 • occnp_diam_response_local_total{appId="16777238",appType="Gx",cmdCode="272",destHost="",destRealm="",direction="out",msgType="CCA-T",origHost="XYZ-IN.us.oracle.com",origRealm="oracle.com",reqDestHost="XYZ-IN.us.oracle.com",reqDestRealm="oracle.com",reqOrigHost="pgw1.oracle.com",reqOrigRealm="oracle.com",responseCode="2001",} 1.0 1633975504473

Table 9-340 occnp_diam_request_local_total (AAR-I)

Field	Details
Description	Number of AAR initial request sent.
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm
Example	occnp_diam_request_local_total{appId="16777236",appType="Rx",cmdCode="265",destHost="",destRealm="oracle.com",direction="in",msgType="AAR-I",origHost="af.oracle.com",origRealm="oracle.com",} 1.0 1634756738735

Table 9-341 occnp_diam_response_local_total (AAA-I)

Field	Details
Description	Number of AAA initial response received.
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm • reqDestHost • reqDestRealm • reqOrigHost • reqOrigRealm • responseCode
Example	occnp_diam_response_local_total{appId="16777236",appType="Rx",cmdCode="265",destHost="",destRealm="",direction="out",msgType="AAA-I",origHost="XYZ-IN.us.oracle.com",origRealm="oracle.com",reqDestHost="",reqDestRealm="oracle.com",reqOrigHost="af.oracle.com",reqOrigRealm="oracle.com",responseCode="5065",} 2.0 1634756738735

Table 9-342 occnp_diam_request_local_total (AAR-U)

Field	Details
Description	Number of AAR modify requests sent.
Type	Counter

Table 9-342 (Cont.) occnp_diam_request_local_total (AAR-U)

Field	Details
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm
Example	occnp_diam_request_local_total{appId="16777236",appType="Rx",cmdCode="265",destHost="pcrf-core",destRealm="oracle.com",direction="in",msgType="AAR-U",origHost="af.oracle.com", origRealm="oracle.com",} 1.0 1635332966264

Table 9-343 occnp_diam_response_local_total (AAA-U)

Field	Details
Description	Number of AAA modify responses received.
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm • reqDestHost • reqDestRealm • reqOrigHost • reqOrigRealm • responseCode
Example	occnp_diam_response_local_total{appId="16777236",appType="Rx",cmdCode="265",destHost="", destRealm="",direction="out",msgType="AAA-U",origHost="pcrf-core", origRealm="oracle.com",reqDestHost="pcrf-core",reqDestRealm="oracle.com",reqOrigHost="af.oracle.com",reqOrigRealm="oracle.com",responseCode="2001",} 1.0 1635332966264

Table 9-344 occnp_diam_request_local_total (RAR Gx)

Field	Details
Description	Number of RAR Gx request received.
Type	Counter

Table 9-344 (Cont.) occnp_diam_request_local_total (RAR Gx)

Field	Details
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm

Table 9-345 occnp_diam_response_local_total (RAA Gx)

Field	Details
Description	Number of RAR Gx response received.
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm • reqDestHost • reqDestRealm • reqOrigHost • reqOrigRealm • responseCode

Table 9-346 occnp_diam_request_local_total (RAR Rx)

Field	Details
Description	Number of RAR Rx request sent.
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm

Table 9-347 occnp_diam_response_local_total (RAA Rx)

Field	Details
Description	Number of RAA Rx response received.
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm • reqDestHost • reqDestRealm • reqOrigHost • reqOrigRealm • responseCode

Table 9-348 occnp_diam_request_local_total (STR)

Field	Details
Description	Number of STR request sent.
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm
Example	occnp_diam_request_local_total{appId="16777236",appType="Rx",cmdCode="275",destHost="XYZ-IN.us.oracle.com",destRealm="oracle.com",direction="in", msgType="STR",origHost="app.oracle.com",origRealm="oracle.com",} 2.0 1635332966264

Table 9-349 occnp_diam_response_local_total (STA)

Field	Details
Description	Number of STA response received.
Type	Counter

Table 9-349 (Cont.) occnp_diam_response_local_total (STA)

Field	Details
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm • reqDestHost • reqDestRealm • reqOrigHost • reqOrigRealm • responseCode
Example	• occnp_diam_response_local_total{appId="16777236",appType="Rx",cmdCode="275",destHost="",destRealm="",direction="out",msgType="STA",origHost="XYZ-IN.us.oracle.com",origRealm="oracle.com",reqDestHost="",reqDestRealm="oracle.com",reqOrigHost="app.oracle.com",reqOrigRealm="oracle.com",responseCode="5002",} 1.0 1635332966264 • occnp_diam_response_local_total{appId="16777236",appType="Rx",cmdCode="275",destHost="",destRealm="",direction="out",msgType="STA",origHost="XYZ-IN.us.oracle.com",origRealm="oracle.com",reqDestHost="",reqDestRealm="oracle.com",reqOrigHost="app.oracle.com",reqOrigRealm="oracle.com",responseCode="5002",} 1.0 1635332966264

Table 9-350 occnp_diam_request_local_total (ASR)

Field	Details
Description	Number of ASR request sent.
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm
Example	occnp_diam_request_local_total{appId="16777236",appType="Rx",cmdCode="274",destHost="af.oracle.com",destRealm="oracle.com",direction="out",msgType="ASR",origHost="XYZ-IN.us.oracle.com",origRealm="oracle.com",} 1.0 1638875554937

Table 9-351 occnp_diam_response_local_total (ASA)

Field	Details
Description	Number of ASA response received.
Type	Counter

Table 9-351 (Cont.) occnp_diam_response_local_total (ASA)

Field	Details
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm • reqDestHost • reqDestRealm • reqOrigHost • reqOrigRealm • responseCode
Example	• occnp_diam_response_local_total{appId="16777236",appType="Rx",cmdCode="274",destHost="",destRealm="",direction="in",msgType="ASA",origHost="app.oracle.com", origRealm="oracle.com",reqDestHost="",reqDestRealm="",reqOrigHost="",reqOrigRealm="",responseCode="2001",} 2.0 1635332966264 • occnp_diam_response_local_total{appId="16777236",appType="Rx",cmdCode="274",destHost="",destRealm="",direction="in",msgType="ASA",origHost="",origRealm="",reqDestHost="af.oracle.com",reqDestRealm="oracle.com",reqOrigHost="XYZ-IN.us.oracle.com",reqOrigRealm="oracle.com",responseCode="timeout",} 1.0 1638875554937

Table 9-352 occnp_diam_request_local_total (CER)

Field	Details
Description	Number of CER request sent.
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm
Example	occnp_diam_request_local_total{appId="0",appType="",cmdCode="257",destHost="",destRealm="",direction="in",msgType="CER",origHost="app.oracle.com", origRealm="oracle.com",} 1.0 1638278712308

Table 9-353 occnp_diam_response_local_total (CEA)

Field	Details
Description	Number of CEA response received.
Type	Counter

Table 9-353 (Cont.) occnp_diam_response_local_total (CEA)

Field	Details
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm • reqDestHost • reqDestRealm • reqOrigHost • reqOrigRealm • responseCode
Example	occnp_diam_response_local_total{appId="0",appType="",cmdCode="257",destHost="",destRealm=="",direction="out",msgType="CEA",origHost="XYZ-IN.us.oracle.com", origRealm="oracle.com",reqDestHost="",reqDestRealm="",reqOrigHost="pgw1.oracle.com",reqOrigRealm="oracle.com",responseCode="2001",} 3.0 1638278712308

Table 9-354 occnp_diam_request_local_total (DWR)

Field	Details
Description	Number of DWR request sent.
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm
Examples	• occnp_diam_request_local_total{appId="0",appType="",cmdCode="280",destHost="",destRealm="",direction="out",msgType="DWR",origHost="XYZ-IN.us.oracle.com", origRealm="oracle.com",} 297.0 1638278712308 • occnp_diam_request_local_total{appId="0",appType="",cmdCode="280",destHost="",destRealm="",direction="in",msgType="DWR",origHost="pgw.oracle.com",origRealm="oracle.com",} 4.0 1640076342932

Table 9-355 occnp_diam_response_local_total (DWA)

Field	Details
Description	Number of DWA response received.
Type	Counter

Table 9-355 (Cont.) occnp_diam_response_local_total (DWA)

Field	Details
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm • reqDestHost • reqDestRealm • reqOrigHost • reqOrigRealm • responseCode
Example	• occnp_diam_response_local_total{appId="0",appType="",cmdCode="280",destHost="",destRealm="",direction="in",msgType="DWA",origHost="pgw1.oracle.com",origRealm="oracle.com", reqDestHost="",reqDestRealm="",reqOrigHost="XYZ-IN.us.oracle.com",reqOrigRealm="oracle.com",responseCode="2001",} 296.0 1638278712308 • occnp_diam_response_local_total{appId="0",appType="",cmdCode="280",destHost="",destRealm="",direction="out",msgType="DWA",origHost="XYZ-IN.us.oracle.com", origRealm="oracle.com",reqDestHost="",reqDestRealm="",reqOrigHost="pgw.oracle.com",reqOrigRealm="oracle.com",responseCode="2001",} 4.0 1640076342932

Table 9-356 occnp_diam_request_local_total (DPR)

Field	Details
Description	Number of DPR request sent.
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm
Example	occnp_diam_request_local_total{appId="0",appType="",cmdCode="282",destHost="",destRealm="",direction="in",msgType="DPR",origHost="pgw1.oracle.com", origRealm="oracle.com",} 1.0 1638278712308

Table 9-357 occnp_diam_response_local_total (DPA)

Field	Details
Description	Number of DPA response received.
Type	Counter

Table 9-357 (Cont.) occnp_diam_response_local_total (DPA)

Field	Details
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm • reqDestHost • reqDestRealm • reqOrigHost • reqOrigRealm • responseCode
Example	occnp_diam_response_local_total{appId="0",appType="",cmdCode="282",destHost="",destRealm="",direction="out",msgType="DPA",origHost="XYZ-IN.us.oracle.com", origRealm="oracle.com",reqDestHost="",reqDestRealm="",reqOrigHost="pgw1.oracle.com",reqOrigRealm="oracle.com",responseCode="2001",} 1.0 1638278712308

Table 9-358 occnp_diam_request_local_total (Timeout Metric RAR)

Field	Details
Description	Number of timeout RAR request sent.
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm
Examples	• occnp_diam_request_local_total{appId="16777238",appType="Gx",cmdCode="258",destHost="pgw1.oracle.com",destRealm="oracle.com",direction="out",msgType="RAR", origHost="XYZ-IN.us.oracle.com",origRealm="oracle.com",} 6.0 1635335195027 • occnp_diam_request_local_total{appId="16777236",appType="Rx",cmdCode="258",destHost="af.oracle.com",destRealm="oracle.com",direction="out",msgType="RAR", origHost="XYZ-IN.us.oracle.com",origRealm="oracle.com",} 1.0 1635425739107 • occnp_diam_request_local_total{appId="16777238",appType="Gx",cmdCode="258",destHost="pgw1.oracle.com",destRealm="oracle.com",direction="out",msgType="RAR",origHost="XYZ-IN.us.oracle.com",origRealm="oracle.com",} 3.0 1638875554937

Table 9-359 occnp_diam_response_local_total (Timeout Metric RAA)

Field	Details
Description	Number of timeout RAA response received.
Type	Counter

Table 9-359 (Cont.) occnp_diam_response_local_total (Timeout Metric RAA)

Field	Details
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm • reqDestHost • reqDestRealm • reqOrigHost • reqOrigRealm • responseCode
Examples	• occnp_diam_response_local_total{appId="16777236",appType="Rx",cmdCode="258",destHost="",destRealm="",direction="in",msgType="RAA",origHost="app.oracle.com",origRealm="oracle.com", reqDestHost="",reqDestRealm="",reqOrigHost="",reqOrigRealm="",responseCode="2001",} 1.0 1635425739107 • occnp_diam_response_local_total{appId="16777238",appType="Gx",cmdCode="258",destHost="",destRealm="",direction="in",msgType="RAA",origHost="pgw1.oracle.com", origRealm="oracle.com",reqDestHost="",reqDestRealm="",reqOrigHost="",reqOrigRealm="",responseCode="2001",} 1.0 1638875554937 • occnp_diam_response_local_total{appId="16777238",appType="Gx",cmdCode="258",destHost="",destRealm="",direction="in",msgType="RAA",origHost="",origRealm="",reqDestHost="pgw1.oracle.com",reqDestRealm="oracle.com",reqOrigHost="XYZ-IN.us.oracle.com",reqOrigRealm="oracle.com",responseCode="timeout",} 2.0 1638875554937

Table 9-360 occnp_diam_request_local_total (Timeout Metric ASR)

Field	Details
Description	Number of timeout ASR request sent.
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm

Table 9-361 occnp_diam_response_local_total (Timeout Metric ASA)

Field	Details
Description	Number of timeout ASA response received.

Table 9-361 (Cont.) occnp_diam_response_local_total (Timeout Metric ASA)

Field	Details
Type	Counter
Dimension	• appId • appType • cmdCode • destHost • destRealm • direction • msgType • origHost • origRealm • reqDestHost • reqDestRealm • reqOrigHost • reqOrigRealm • responseCode

Bulwark Metrics

Table 9-362 occnp_http_bulwark_lock_request_total

Field	Details
Description	Number of bulwark lock requests sent.
Type	Counter
Dimension	• application • instanceId • operationType • msgType
Example	occnp_http_bulwark_lock_request_total{application="occnp-pcrf-core",instanceId="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",msgType="create",operationType="bulwark_lock",} 2.0

Table 9-363 occnp_http_bulwark_lock_response_total

Field	Details
Description	Number of bulwark lock response received.
Type	Counter
Dimension	• application • instanceId • operationType • responseCode • msgType
Example	occnp_http_bulwark_lock_response_total{application="occnp-pcrf-core",instanceId="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",msgType="create",operationType="bulwark_lock",responseCode="201",} 2.0

Table 9-364 occnp_http_bulwark_unlock_request_total

Field	Details
Description	Number of bulwark unlock requests sent.
Type	Counter
Dimension	• application • instanceId • operationType • msgType
Example	occnp_http_bulwark_unlock_request_total{application="occnp-pcrf-core",instanceId="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",msgType="create",operationType="bulwark_unlock",} 2.0

Table 9-365 occnp_http_bulwark_unlock_response_total

Field	Details
Description	Number of bulwark unlock response received.
Type	Counter
Dimension	• application • instanceId • operationType • responseCode • msgType
Example	occnp_http_bulwark_unlock_response_total{application="occnp-pcrf-core",instanceId="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",msgType="create",operationType="bulwark_unlock",responseCode="204",} 2.0

Binding and PRE Metrics

Table 9-366 http_out_conn_request_total

Field	Details
Description	Number of connection requests sent.
Type	Counter
Dimension	• application • operationType • serviceResource
Example	http_out_conn_request_total{application="pcrf-core",operationType="DELETE",serviceResource="Binding",} 28.0

Table 9-367 http_out_conn_response_total

Field	Details
Description	Number of connection response received.
Type	Counter

Table 9-367 (Cont.) http_out_conn_response_total

Field	Details
Dimension	• application • operationType • responseCode • serviceResource
Example	http_out_conn_response_total{application="pcrf-core",operationType="POST",responseCode="201",serviceResource="Binding",} 33.0

Table 9-368 pcrf_core_binding_ex_total

Field	Details
Description	Accounts for any exceptions that occurred during API calls toward Binding service.
Type	Counter
Dimension	• application • operationType • exception_type
Example	**General Metrics** pcrf_core_binding_ex_total{application="occnp-pcrf-core",exception_type="java.util.concurrent.TimeoutException",operationType="POST",} 2.0 dimensions: **Binding and Pre Metrics** pcrf_core_binding_ex_total{application="occnp-pcrf-core",exception_type="java.util.concurrent.TimeoutException",operationType="POST",} 2.0

Webclient Metrics

Table 9-369 occnp_webclient_sum_total

Field	Details
Description	Sum of webclient requests.
Type	Counter
Dimension	• method • status_code • url
Example	occnp_webclient_sum_total{method="POST",status_code="201",url="/binding/v1/dependentContextBinding/context-owner/PCRF-CORE",} 201.0

Table 9-370 occnp_webclient_count_total

Field	Details
Description	Number of webclient requests.
Type	Counter
Dimension	• method • status_code • url

Table 9-370 (Cont.) occnp_webclient_count_total

Field	Details
Example	occnp_webclient_count_total{method="POST",status_code="201",url="/binding/v1/dependentContextBinding/context-owner/PCRF-CORE",} 6.0

JDBC Metrics

Table 9-371 occnp_jdbc_operation_count_total

Field	Details
Description	Number of jdbc operations.
Type	Counter
Dimension	• method • state • table
Example	occnp_jdbc_operation_count_total{method="removeSQL",state="SUCCESS",table="rxsession",} 6.0

Table 9-372 occnp_jdbc_operation_sum_total

Field	Details
Description	Sum of jdbc operations.
Type	Counter
Dimension	• method • state • table
Example	occnp_jdbc_operation_sum_total{method="removeSQL",state="SUCCESS",table="rxsession",} 25.0

Session Viewer Metrics

Table 9-373 session_oam_request_total

Field	Details
Description	Total number of oam requests sent.
Type	Counter
Dimension	• application • operationType • sessionType
Example	session_oam_request_total{application="pcf_smservice",operationType="terminate",sessionType="SM"} 2.0

Table 9-374 session_oam_response_total

Field	Details
Description	Total number of oam response received.
Type	Counter

Table 9-374 (Cont.) session_oam_response_total

Field	Details
Dimension	• application • operationType • responseCode • sessionType
Example	session_oam_response_total{application="pcf_smservice",operationType="terminate",responseCode="5xx",sessionType="SM"} 1.0

Jetty HTTP client request timing Metrics

Table 9-375 jetty_client_requests_seconds_max

Field	Details
Description	Number of jetty client requests.
Type	Counter
Dimension	• exception • host • method • outcome • status • uri
Example	jetty_client_requests_seconds_max{exception="None",host="pdincredpolicy-occnp-binding",method="DELETE",outcome="SUCCESS",status="204",uri="",} 0.0

Table 9-376 jetty_client_requests_seconds_sum

Field	Details
Description	Sum of jetty client requests.
Type	Counter
Dimension	• exception • host • method • outcome • status • uri
Example	jetty_client_requests_seconds_sum{exception="None",host="pdincredpolicy-occnp-binding",method="DELETE",outcome="SUCCESS",status="204",uri="",} 0.831213657

Table 9-377 jetty_client_requests_sum_total

Field	Details
Description	Sum of jetty client requests.
Type	Counter
Dimension	• method • status_code • url

Table 9-377 (Cont.) jetty_client_requests_sum_total

Field	Details
Example	jetty_client_requests_sum_total{method="POST",status_code="201",url="/binding/v1/dependentContextBinding/context-owner/PCRF-CORE",} 199.0

Table 9-378 jetty_client_request_size_sum_total

Field	Details
Description	Sum of jetty client request size.
Type	Counter
Dimension	• method • status_code • url
Example	jetty_client_request_size_sum_total{method="DELETE",status_code="204",url="/binding/v1/dependentContextBinding/dependentContextId",} 0.0

Table 9-379 jetty_client_requests_count_total

Field	Details
Description	Number of jetty client requests.
Type	Counter
Dimension	• method • status_code • url
Example	jetty_client_requests_count_total{method="POST",status_code="201",url="/binding/v1/dependentContextBinding/context-owner/PCRF-CORE",} 6.0

Cgroup Metrics

Table 9-380 cgroup_cpu_nanoseconds

Field	Details
Description	Reports the total CPU time (in nanoseconds) on each CPU core for all the tasks in the cgroup.
Type	Gauge
Dimension	NA
Example	cgroup_cpu_nanoseconds 2.1782821080274e+013

Table 9-381 cgroup_memory_bytes

Field	Details
Description	Reports the memory usage.
Type	Gauge
Dimension	NA
Example	cgroup_memory_bytes 1.31289088e+08

Startup Probe Metrics

Table 9-382 config_client_audit_complete

Field	Details
Description	Reports the audit completion of config client.
Type	Gauge
Dimension	auditor
Example	config_client_audit_complete_total{auditor="configClientAuditor",} 7768.0

Database Data Encoding Metrics

Table 9-383 occnp_data_encoding_total

Field	Details
Description	PCRF Core gxsession, rxsession and sdsession total of encoded data.
Type	Counter
Dimension	• db (occnp_pcrf_core) • table (gxsession, rxsession, sdsession) • decodingVer (mapping version number used for decoding)
Example	

Table 9-384 occnp_data_decoding_total

Field	Details
Description	PCRF Core gxsession, rxsession and sdsession total of decoded data.
Type	Counter
Dimension	• db (occnp_pcrf_core) • table (gxsession, rxsession, sdsession) • decodingVer (mapping version number used for decoding)
Example	

Table 9-385 occnp_data_decoding_fail_total

Field	Details
Description	PCRF Core gxsession, rxsession and sdsession total of decoding failures
Type	Counter
Dimension	• db (occnp_pcrf_core) • table (gxsession, rxsession, sdsession) • decodingVer (mapping version number used for decoding) • errorCause (json_processing_exception, encode_version_null, empty_key_value_map)
Example	

Table 9-386 occnp_data_encoding_size_before_total

Field	Details
Description	PCRF Core gxsession, rxsession and sdsession total size before encoding.

Table 9-386 (Cont.) occnp_data_encoding_size_before_total

Field	Details
Type	Counter
Dimension	• db (occnp_pcrf_core) • table (gxsession, rxsession, sdsession)
Example	

Table 9-387 occnp_data_encoding_size_after_total

Field	Details
Description	PCRF Core gxsession, rxsession and sdsession total size after encoding.
Type	Counter
Dimension	• db (occnp_pcrf_core) • table (gxsession, rxsession, sdsession)
Example	

9.23 Late Arrival Requests and Collision Detection Metrics

The following table describes the late arrival requests and collision detection metrics and their respective dimensions:

Table 9-388 ocpm_late_arrival_rejection_total

Field	Details
Description	Metric that increments when AM, UE, or SM Policy Association late arriving requests are detected and rejected.
Type	Counter
Dimensions	• OPERATION_TYPE • TGPP_SERVICE • CAUSE

Table 9-389 ocpm_collision_detection_total

Field	Details
Description	Metric that increments when AM, UE, or SM Policy Association duplicate records are detected and deleted or rejected.
Type	Counter
Dimensions	• SERVICE_NAME • OPERATION_TYPE • CAUSE

For more information about Dimensions, see CNC Policy Metrics.

9.24 Notifier Metrics

The following table describes the Notifier service metrics and their respective dimensions:

Table 9-390 http_notification_request_total

Field	Details
Description	This metric tracks the total number of requests towards a given destination.
Type	Counter
Dimensions	notificationEndpoint = IP or FQDN of endpoint version: http1 or http2 https = true or false contentType = <From Content-type header> method = <HTTP method for notification>

Table 9-391 http_notification_response_total

Field	Details
Description	This metric tracks the total number of responses from a given destination.
Type	Counter
Dimensions	notificationEndpoint = IP or FQDN of endpoint version: http1 or http2 https = true or false contentType = <From Content-type header> method = <HTTP method for notification> responseCode = Response code

Table 9-392 http_notification_request_timeout_total

Field	Details
Description	This metric indicates the request timeout period. That is, the maximum time for which the request can last, after which it is considered as timeout.
Type	Counter
Dimensions	notificationEndpoint = IP or FQDN of endpoint version: http11 https = false contentType = <From Content-type header> method = <HTTP method for notification>

Table 9-393 http_in_conn_request_total

Field	Details
Description	This metric indicates the request coming from PRE
Type	Counter
Dimensions	• OperationType : NOTIFY • servicenameNon3gpp : notifier
Example	occnp_http_in_conn_request_total{operationType="NOTIFY",servicenameNon3gpp="notifier",} 1.0

Table 9-394 http_in_conn_response_total

Field	Details
Description	This metric indicates the response sent to PRE
Type	Counter
Dimensions	• OperationType : NOTIFY • responseCode • servicenameNon3gpp : notifier • cause:SMPP is not enabled
Example	occnp_http_in_conn_response_total{operationType="NOTIFY",responseCode="204 NO_CONTENT",servicenameNon3gpp="notifier",} 1.0

Table 9-395 smpp_request_total

Field	Details
Description	This metric indicates the requests sent towards external endpoint such as SMS Gateway and the delivery receipts received.
Type	Counter
Dimensions	• notificationServer: (sms gateway name) • smscName: name • direction:(in/out) • requestType: bind/ unbind/submit_sm/deliver_sm • deliveryMethod: SAR/MESSAGE_PAYLOAD
Examples	• occnp_smpp_request_total{direction="out",notificationServer="gw1",requestType="submit_sm",smscName="host3",} 1.0 • occnp_smpp_request_total{direction="out",notificationServer="",requestType="unbind",smscName="host3",} 1.0 • ccnp_smpp_request_total{direction="out",notificationServer="",requestType="bind",smscName="host3",} 1.0 • occnp_smpp_request_total{direction="in",notificationServer="",requestType="deliver_sm",smscName="host3",} 1.0 • occnp_smpp_request_total{direction="out",notificationServer="sms-gw-1",requestType="submit_sm",smscName="host3",} 1.0

Table 9-396 smpp_response_total

Field	Details
Description	Indicates the number of respoinses received from an external endpoint
Type	Counter
Dimensions	• notificationServer: (sms gateway name) • smscName: hostname • response : (SUCCESS, FAILURE, UNKNOWN) • requestType: bind_resp/unbind_resp/submit_sm_resp/deliver_sm_resp • latency: • cause: (ENROUTE, DELIVRD, EXPIRED,DELETED,UNDELIV,ACCEPTD, UNKNOWN,REJECTD)

Table 9-396 (Cont.) smpp_response_total

Field	Details
Examples	• occnp_smpp_response_total{cause="",direction="in",latency="0",notificationServer="sms-gw-1",requestType="submit_sm_resp",response="SUCCESS",smscName="host3",} 1.0 • occnp_smpp_response_total{cause="",direction="in",latency="9",notificationServer="",requestType="bind_resp",response="SUCCESS",smscName="host3",} 1.0 • occnp_smpp_response_total{cause="",direction="in",latency="0",notificationServer="",requestType="unbind_resp",response="SUCCESS",smscName="host3",} 1.0 • occnp_smpp_response_total{cause="DELIVRD",direction="out",latency="0",notificationServer="",requestType="deliver_sm_resp",response="SUCCESS",smscName="host3",} 1.0 • occnp_smpp_response_total{cause="",direction="in",latency="9",notificationServer="gw1",requestType="submit_sm_resp",response="SUCCESS",smscName="host3",} 1.0

Table 9-397 active_smsc_conn_count

Field	Details
Description	captures total number of active connection with smsc hosts.
Type	Gauge
Dimensions	smscName: name
Example	occnp_active_smsc_conn_count{smscName="host3",} 1.0

Table 9-398 topic_version

Field	Details
Description	Describes the current applied version of a given topic (mentioned in dimension topic_name) into the pod.
Type	Gauge
Dimensions	• Service Name • Pod Name

Histograms

```
# HELP occnp_smpp_message_processing_latency_seconds
# TYPE occnp_smpp_message_processing_latency_seconds histogram
occnp_smpp_message_processing_latency_seconds{requestType="bind",quantile="0.5
",} 1.207959552
occnp_smpp_message_processing_latency_seconds{requestType="bind",quantile="0.9
",} 2.483027968
occnp_smpp_message_processing_latency_seconds{requestType="bind",quantile="0.9
5",} 2.483027968
occnp_smpp_message_processing_latency_seconds{requestType="bind",quantile="0.9
9",} 2.483027968
occnp_smpp_message_processing_latency_seconds_bucket{requestType="bind",le="0.
01",} 0.0
```

```
occnp_smpp_message_processing_latency_seconds_bucket{requestType="bind",le="0.
02",} 0.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="bind",le="0.
04",} 0.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="bind",le="0.
08",} 2.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="bind",le="0.
1",} 2.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="bind",le="0.
2",} 2.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="bind",le="0.
5",} 2.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="bind",le="1.
0",} 70.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="bind",le="5.
0",} 73.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="bind",le="+I
nf",} 73.0
occnp_smpp_message_processing_latency_seconds_count{requestType="bind",} 73.0
occnp_smpp_message_processing_latency_seconds_sum{requestType="bind",} 55.069

occnp_smpp_message_processing_latency_seconds{requestType="submit_sm",quantile
="0.5",} 0.014680064
occnp_smpp_message_processing_latency_seconds{requestType="submit_sm",quantile
="0.9",} 0.01835008
occnp_smpp_message_processing_latency_seconds{requestType="submit_sm",quantile
="0.95",} 0.01835008
occnp_smpp_message_processing_latency_seconds{requestType="submit_sm",quantile
="0.99",} 0.01835008
occnp_smpp_message_processing_latency_seconds_bucket{requestType="submit_sm",l
e="0.01",} 3.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="submit_sm",l
e="0.02",} 6.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="submit_sm",l
e="0.04",} 6.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="submit_sm",l
e="0.08",} 7.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="submit_sm",l
e="0.1",} 7.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="submit_sm",l
e="0.2",} 7.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="submit_sm",l
e="0.5",} 7.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="submit_sm",l
e="1.0",} 7.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="submit_sm",l
e="5.0",} 7.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="submit_sm",l
e="+Inf",} 8.0
occnp_smpp_message_processing_latency_seconds_count{requestType="submit_sm",}
8.0
occnp_smpp_message_processing_latency_seconds_sum{requestType="submit_sm",}
6.084

occnp_smpp_message_processing_latency_seconds{requestType="unbind",quantile="0
.5",} 0.017825792
```

```
occnp_smpp_message_processing_latency_seconds{requestType="unbind",quantile="0
.9",} 0.017825792
occnp_smpp_message_processing_latency_seconds{requestType="unbind",quantile="0
.95",} 0.017825792
occnp_smpp_message_processing_latency_seconds{requestType="unbind",quantile="0
.99",} 0.017825792
occnp_smpp_message_processing_latency_seconds_bucket{requestType="unbind",le="
0.01",} 0.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="unbind",le="
0.02",} 1.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="unbind",le="
0.04",} 1.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="unbind",le="
0.08",} 1.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="unbind",le="
0.1",} 1.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="unbind",le="
0.2",} 1.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="unbind",le="
0.5",} 1.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="unbind",le="
1.0",} 1.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="unbind",le="
5.0",} 1.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="unbind",le="
+Inf",} 1.0
occnp_smpp_message_processing_latency_seconds_count{requestType="unbind",} 1.0
occnp_smpp_message_processing_latency_seconds_sum{requestType="unbind",} 0.018


occnp_smpp_message_processing_latency_seconds{requestType="deliver_sm",quantil
e="0.5",} 0.009404416
occnp_smpp_message_processing_latency_seconds{requestType="deliver_sm",quantil
e="0.9",} 0.015171584
occnp_smpp_message_processing_latency_seconds{requestType="deliver_sm",quantil
e="0.95",} 0.015171584
occnp_smpp_message_processing_latency_seconds{requestType="deliver_sm",quantil
e="0.99",} 0.015171584
occnp_smpp_message_processing_latency_seconds_bucket{requestType="deliver_sm",
le="0.01",} 3.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="deliver_sm",
le="0.02",} 4.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="deliver_sm",
le="0.04",} 4.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="deliver_sm",
le="0.08",} 4.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="deliver_sm",
le="0.1",} 4.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="deliver_sm",
le="0.2",} 4.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="deliver_sm",
le="0.5",} 4.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="deliver_sm",
le="1.0",} 4.0
occnp_smpp_message_processing_latency_seconds_bucket{requestType="deliver_sm",
le="5.0",} 4.0
```

```
occnp_smpp_message_processing_latency_seconds_bucket{requestType="deliver_sm",
le="+Inf",} 4.0
occnp_smpp_message_processing_latency_seconds_count{requestType="deliver_sm",}
 4.0
occnp_smpp_message_processing_latency_seconds_sum{requestType="deliver_sm",}
0.035


# HELP occnp_smpp_message_processing_latency_seconds_max
# TYPE occnp_smpp_message_processing_latency_seconds_max gauge
occnp_smpp_message_processing_latency_seconds_max{requestType="bind",} 0.803
occnp_smpp_message_processing_latency_seconds_max{requestType="unbind",} 0.018
occnp_smpp_message_processing_latency_seconds_max{requestType="submit_sm",}
0.025
occnp_smpp_message_processing_latency_seconds_max{requestType="deliver_sm",}
0.015
```

9.25 Usage Monitoring Metrics

The following table describes the Usage Monitoring service metrics and their respective dimensions:

Table 9-399 usage_mon_ingress_request

Field	Details
Description	This metric is incremented when Usage Monitoring service receives an incoming HTTP request message.
Type	Counter
Dimensions	requestor_nf, operation_type, dnn, snssai
Example	

Table 9-400 usage_mon_ingress_response

Field	Details
Description	This metric is incremented when Usage Monitoring service responds to an incoming HTTP request message.
Type	Counter
Dimensions	requestor_nf, operation_type, dnn, snssai, response_code, latency
Example	

Table 9-401 usage_mon_profile_activated

Field	Details
Description	This metric is incremented when UM Data Limit Profile is activated.
Type	Counter
Dimensions	limit_id
Example	

Table 9-402 usage_mon_profile_selected

Field	Details
Description	This metric is incremented when UM Data Limit Profile is selected.
Type	Counter
Dimensions	limit_id
Example	

Table 9-403 usage_mon_grant_success

Field	Details
Description	This metric is incremented when a grant is successfully allocated from a given UM Data Limit Profile.
Type	Counter
Dimensions	limit_id
Example	

Table 9-404 usage_mon_grant_failure

Field	Details
Description	This metric is incremented when a grant could not be allocated from a given UM Data Limit Profile.
Type	Counter
Dimensions	limit_id
Example	

Table 9-405 usage_mon_usage_reported

Field	Details
Description	This metric is incremented when usage report is processed for a given UM Data Limit Profile.
Type	Counter
Dimensions	monitoring_key
Example	

Table 9-406 usage_mon_usage_threshold

Field	Details
Description	This metric is incremented when usage usage crosses a particular threshold level.
Type	Counter
Dimensions	limit_id, level
Example	

Table 9-407 usage_mon_usage_reset

Field	Details
Description	This metric is incremented when usage is reset according to the reset time in a given UM Data Limit Profile.
Type	Counter
Dimensions	limit_id
Example	

Table 9-408 usage_mon_egress_pds_request

Field	Details
Description	This metric is incremented when Usage Monitoring service sends a request to PDS service.
Type	Counter
Dimensions	operation_type
Example	

Table 9-409 usage_mon_egress_pds_response

Field	Details
Description	This metric is incremented when Usage Monitoring service receives a response from PDS service.
Type	Counter
Dimensions	operation_type, response_code, latency
Example	

Table 9-410 session_oam_request_total

Field	Details
Description	Number of session requests for Usage Monitoring Service
Type	Request
Dimensions	• Response Code • Operation Type • Session Type
Example	

Table 9-411 session_oam_response_total

Field	Details
Description	Number of session responses for Usage Monitoring Service
Type	Response
Dimensions	• Response Code • Operation Type • Session Type
Example	

Table 9-412 topic_version

Field	Details
Description	Describes the current applied version of a given topic (mentioned in dimension topic_name) into the pod.
Type	Gauge
Dimensions	• Service Name • Pod Name
Example	

Table 9-413 occnp_db_overall_processing_time

Field	Details
Description	Indicates the status of the failure or success operation of data compression on UmDataContext table for Usage Monitoring.
Type	Gauge
Dimensions	• Table • Method • Status • compressionScheme
Example	

Table 9-414 usage_mon_context_found

Field	Details
Description	This metric is incremented when session lookup happens on the main table or sliced tables for old sessions of the subscribers when table slicing is enabled.
Type	Counter
Dimensions	• table_count • operation_type • table_number
Example	

Table 9-415 occnp_um_audit_deleted_maxttl

Field	Details
Description	This metric is incremented when MaxTTL is reached for a enforcement session.
Type	Counter
Dimensions	siteId
Example	

Table 9-416 occnp_um_audit_query_sent

Field	Details
Description	This metric is incremented when the Usage Monitoring service sends a query to PCRF Core to check session staleness.

Table 9-416 (Cont.) occnp_um_audit_query_sent

Field	Details
Type	Counter
Dimensions	siteId
Example	

Table 9-417 occnp_um_audit_session_found

Field	Details
Description	This metric is incremented when Usage Monitoring receives 2xx response code from PCRF Core for a session GET request.
Type	Counter
Dimensions	siteId
Example	

Table 9-418 occnp_um_audit_session_not_found

Field	Details
Description	This metric is incremented when Usage Monitoring receives 404 response code from PCRF Core for session GET request.
Type	Counter
Dimensions	siteId
Example	

Table 9-419 occnp_um_audit_query_error_response

Field	Details
Description	This metric is incremented when Usage Monitoring receives 3xx, 4xx (except 404) and 5xx response code from PCRF Core for a session GET request.
Type	Counter
Dimensions	• siteId • response_code
Example	

Table 9-420 occnp_um_audit_query_exception

Field	Details
Description	This metric is incremented when Usage Monitoring service ends up with an exception while sending GET request to PCRF Core service.
Type	Counter
Dimensions	siteId
Example	

Table 9-421 occnp_usage_mon_rollover_capped_total

Field	Details
Description	This metric is incremented when rollover reaches maximum data limit cap.
Type	String
Dimensions	• limit_id • hostname
Example	occnp_usage_mon_rollover_capped_total{hostname="my-cnpolicy-usage-mon-6ff7947d8f-rcrxs",limit_id="mk1-roll-over-def-fa74b",} 1.0 occnp_usage_mon_rollover_capped_total {hostname="ocpcf-usage-mon-b564d6cf8-d5d79",limit_id="mk1-roll-over-def-18e63",}1.0 occnp_usage_mon_rollover_capped_total{hostname="ocpcf-usage-mon-b564d6cf8-d5d79",limit_id="mk1-roll-over-def-ef270",}1.0 occnp_usage_mon_rollover_capped_total{hostname="ocpcf-usage-mon-b564d6cf8-d5d79",limit_id="mk1-roll-over-def-63e0c",}1.0 occnp_usage_mon_rollover_capped_total{hostname="ocpcf-usage-mon-b564d6cf8-d5d79",limit_id="mk1-roll-over-def-10685",}1.0

9.26 Bulwark Metrics

The following table describes the Bulwark service metrics and their respective dimensions:

Table 9-422 lock_request_total

Field	Details
Description	Total number of lock requests received at Bulwark per service or pod.
Type	Counter
Dimensions	• application • podName • requestType (releaseLock, acquireLock) • serviceName
Examples	• lock_request_total{application="bulwark",podName="abc-cnv-rls-bulwark-856556f674-twkfn",requestType="releaseLock",serviceName="serviceId", } 2.0 • lock_request_total{application="bulwark",podName="abc-cnv-rls-bulwark-856556f674-twkfn",requestType="acquireLock",serviceName="serviceId", } 2.0

Table 9-423 lock_response_total

Field	Details
Description	Total number of lock responses sent by Bulwark per service or pod.
Type	Counter
Dimensions	• RequestType (acquire) • ResponseType (success, failure) • ResponseStatus • consumerServiceName • totalLockAttempts • retryOnAlreadyLocked (true, false)

Table 9-423 (Cont.) lock_response_total

Field	Details
Examples	• lock_response_total{application="bulwark",podName="abc-cnv-rls-bulwark-856556f674-twkfn",requestType="acquireLock",responseStatus="201" ,responseType="success" ,serviceName="SM-Service", } 2.0 • lock_response_total{application="bulwark",podName="abc-cnv-rls-bulwark-856556f674-twkfn",requestType="acquireLock",responseStatus="423" ,responseType="failure" ,serviceName="SM-Service", } 2.0 • lock_response_total{application="bulwark",podName="abc-cnv-rls-bulwark-856556f674-twkfn",requestType="releaseLock",responseStatus="204" ,responseType="success" ,serviceName="serviceId", } 2.0

Table 9-424 lock_collision_total

Field	Details
Description	Tracks the total count of all the collisions occurred for single/multi lock request failure.
Type	Counter
Dimensions	• bulwarkPodName • consumerServiceName • lockRequestKeysTotal
Examples	lock_collision_total{application="bulwark",noOfKeysPerRequest="1",podName="pcf-bulwark-776df8bb5c-6g2f6",serviceName="policyds",} 80.0 lock_collision_total{application="bulwark",noOfKeysPerRequest="2",podName="pcf-bulwark-776df8bb5c-6g2f6",serviceName="occnp_pcf_sm",} 8.0 lock_collision_total{application="bulwark",noOfKeysPerRequest="1",podName="pcf-bulwark-776df8bb5c-6g2f6",serviceName="occnp_pcf_sm",} 92.0

Table 9-425 coherence_callback_operation_total

Field	Details
Description	Tracks the total count all the register/deregister requests towards coherence.
Type	Counter
Dimensions	• bulwarkPodName • opType (Registration, Deregistration) • Status (SUCCESS, FAILED)
Examples	coherence_callback_operation_total{application="bulwark",opStatus="success",opType="Deregistration",podName="test-bulwark-5f44788c69-qjdhd",} 0 coherence_callback_operation_total{application="bulwark",opStatus="success",opType="Registration",podName="test-bulwark-5f44788c69-qjdhd",} 0

Table 9-426 coherence_callback_response_total

Field	Details
Description	Tracks the total coherence callback responses from coherence to bulwark
Type	Counter

Table 9-426 (Cont.) coherence_callback_response_total

Field	Details
Dimensions	bulwarkPodName
Examples	coherence_callback_response_total{application="bulwark",podName="test-bulwark-8f8b9bc67-dpf85",} 0

For more information about Dimensions, see CNC Policy Metrics.

9.27 CHF Metrics

The following table describes the CHF service metrics and their respective dimensions:

Table 9-427 ocpm_chf_tracking_request_total

Field	Details
Description	Total number of CHF tracking requests.
Type	Counter
Dimensions	• HostName • application • nf_instance_id • nf_name • operation_type • service_resource • service_version • servicename_3gpp
Examples	• ocpm_chf_tracking_request_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-666666666667",nf_name="chf.oracle.com",operation_type="unsubscribe",service_resource="subscriptions",service_version="v1",servicename_3gpp="nchf-spendinglimitcontrol",} 0.0 ; Type-Counter • ocpm_chf_tracking_request_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-666666666667",nf_name="chf.oracle.com",operation_type="put",service_resource="subscriptions",service_version="v1",servicename_3gpp="nchf-spendinglimitcontrol",} 0.0 ; Type-Counter • ocpm_chf_tracking_request_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-666666666667",nf_name="chf.oracle.com",operation_type="subscribe",service_resource="subscriptions",service_version="v1",servicename_3gpp="nchf-spendinglimitcontrol",} 1.0 ; Type-Counter

Table 9-428 ocpm_chf_tracking_response_total

Field	Details
Description	Total number of CHF tracking response.
Type	Counter

Table 9-428 (Cont.) ocpm_chf_tracking_response_total

Field	Details
Dimensions	• HostName • application • nf_instance_id • nf_name • operation_type • service_resource • service_version • servicename_3gpp
Examples	• ocpm_chf_tracking_response_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-666666666667",nf_name="chf.oracle.com",operation_type="subscribe",response_code="5xx",service_resource="subscriptions",service_version="v1",servicename_3gpp="nchf-spendinglimitcontrol",} 0.0 ; Type-Counter • ocpm_chf_tracking_response_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-666666666667",nf_name="chf.oracle.com",operation_type="put",response_code="4xx",service_resource="subscriptions",service_version="v1",servicename_3gpp="nchf-spendinglimitcontrol",} 0.0 ; Type-Counter • ocpm_chf_tracking_response_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-666666666667",nf_name="chf.oracle.com",operation_type="put",response_code="1xx",service_resource="subscriptions",service_version="v1",servicename_3gpp="nchf-spendinglimitcontrol",} 0.0 ; Type-Counter • ocpm_chf_tracking_response_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-666666666667",nf_name="chf.oracle.com",operation_type="unsubscribe",response_code="4xx",service_resource="subscriptions",service_version="v1",servicename_3gpp="nchf-spendinglimitcontrol",} 0.0 ; Type-Counter • ocpm_chf_tracking_response_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-666666666667",nf_name="chf.oracle.com",operation_type="unsubscribe",response_code="2xx",service_resource="subscriptions",service_version="v1",servicename_3gpp="nchf-spendinglimitcontrol",} 0.0 ; Type-Counter

9.28 UDR Metrics

The following table describes the UDR service metrics and their respective dimensions:

Table 9-429 ocpm_udr_tracking_request_total

Field	Details
Description	Total number of UDR tracking request.
Type	Counter

Table 9-429 (Cont.) ocpm_udr_tracking_request_total

Field	Details
Dimensions	• HostName • application • nf_instance_id • operation_type • service_resource • service_subresource • service_version • servicename_3gpp
Examples	• ocpm_udr_tracking_request_total{HostName="",application="pcf_userservice",hostname="pcf_userservice",nf_instance_id="3db4b97c-04dc-4aff-ab20-2e156dbd02d2",nf_name="fqdn",operation_type="get",service_resource="policy-data",service_subresource="operator-specific-data",service_version="v1",servicename_3gpp="nudr-dr",} 0.0 ; Type-Counter • ocpm_udr_tracking_request_total{HostName="",application="pcf_userservice",hostname="hummingbirds-pcf190rc4-occnp-udr-connector-7d648f5765-zrwc9",nf_name="udr.oracle.com",nf_instance_id="fe7d992b-0541-4c7d-ab84-6d70b1babc1",operation_type="put",service_resource="policy-data",service_subresource="subs-to-notify",service_version="v1",servicename_3gpp="nudr-dr",} 0.0 ; Type-Counter • ocpm_udr_tracking_request_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-c6d70b1babc1",operation_type="subscribe",service_resource="policy-data",service_subresource="subs-to-notify",service_version="v1",servicename_3gpp="nudr-dr",} 1.0 ; Type-Counter • ocpm_udr_tracking_request_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-c6d70b1babc1",operation_type="unsubscribe",service_resource="policy-data",service_subresource="sm-data",service_version="v1",servicename_3gpp="nudr-dr",} 0.0 ; Type-Counter • ocpm_udr_tracking_request_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-c6d70b1babc1",operation_type="unsubscribe",service_resource="policy-data",service_subresource="subs-to-notify",service_version="v1",servicename_3gpp="nudr-dr",} 0.0 ; Type-Counter • ocpm_udr_tracking_request_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-c6d70b1babc1",operation_type="get",service_resource="policy-data",service_subresource="ue-policy-set",service_version="v1",servicename_3gpp="nudr-dr",} 0.0 ; Type-Counter

Table 9-430 ocpm_udr_tracking_response_total

Field	Details
Description	Total number of UDR tracking response.
Type	Counter

Table 9-430 (Cont.) ocpm_udr_tracking_response_total

Field	Details
Dimensions	• HostName • application • nf_instance_id • operation_type • service_resource • service_subresource • service_version • servicename_3gpp
Examples	• ocpm_udr_tracking_response_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-c6d70b1babc1",nf_name="udr.oracle.com",operation_type="subscribe",response_code="2xx",service_resource="policy-data",service_subresource="",service_version="v1",servicename_3gpp="nudr-dr",} 0.0 ; Type-Counter • ocpm_udr_tracking_response_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-c6d70b1babc1",nf_name="udr.oracle.com",operation_type="unsubscribe",response_code="5xx",service_resource="policy-data",service_subresource="am-data",service_version="v1",servicename_3gpp="nudr-dr",} 0.0 ; Type-Counter • ocpm_udr_tracking_response_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-c6d70b1babc1",nf_name="udr.oracle.com",operation_type="unsubscribe",response_code="1xx",service_resource="policy-data",service_subresource="subs-to-notify",service_version="v1",servicename_3gpp="nudr-dr",} 1.0 ; Type-Counter • ocpm_udr_tracking_response_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-c6d70b1babc1",nf_name="udr.oracle.com",operation_type="put",response_code="1xx",service_resource="policy-data",service_subresource="sm-data",service_version="v1",servicename_3gpp="nudr-dr",} 0.0 ; Type-Counter • ocpm_udr_tracking_response_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-c6d70b1babc1",nf_name="udr.oracle.com",operation_type="subscribe",response_code="3xx",service_resource="policy-data",service_subresource="subs-to-notify",service_version="v1",servicename_3gpp="nudr-dr",} 1.0 ; Type-Counter • ocpm_udr_tracking_response_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-c6d70b1babc1",nf_name="udr.oracle.com",operation_type="get",response_code="2xx",service_resource="policy-data",service_subresource="",service_version="v1",servicename_3gpp="nudr-dr",} 0.0 ; Type-Counter • ocpm_udr_tracking_response_total{HostName="",application="pcf_userservice",nf_instance_id="fe7d992b-0541-4c7d-ab84-c6d70b1babc1",nf_name="udr.oracle.com",operation_type="patch",response_code="2xx",service_resource="policy-data",service_subresource="am-data",service_version="v1",servicename_3gpp="nudr-dr",} 1.0 ; Type-Counter • ocpm_udr_tracking_response_total{HostName="",application="pcf_userservice",hostname="pcf_userservice",nf_instance_id="3db4b97c-04dc-4aff-ab20-2e156dbd02d2",nf_name="fqdn",operation_type="get",response_code="2xx",service_resource="policy-data",service_subresource="operator-specific-data",service_version="v1",servicename_3gpp="nudr-dr",} 0.0 ; Type-Counter

Table 9-431 ocpm_nfDiscovery_request_total

Field	Details
Description	PCF measures the total number of successful/failed discovery query of UDR using "group-id-list".
Type	Counter
Dimensions	queryParameters="service-names,group-id-list,target-nf-type
Example	ocpm_nfDiscovery_request_total{app_kubernetes_io_instance="test", app_kubernetes_io_managed_by="Helm", app_kubernetes_io_name="user-service", app_kubernetes_io_part_of="occnp", app_kubernetes_io_version="1.0.0", application="occnp", container="user-service", endpoint="cnc-metrics", engVersion="23.4.0-ocngf-incredibles", exported_application="pcf_userservice", helm_sh_chart="user-service-23.4.0-ocngf-incredibles", hostname="test-occnp-udr-connector-868c766947-fr7zx", instance="10.233.93.4:9000", io_kompose_service="test-occnp-udr-connector", job="occne-infra/occne-nf-cnc-podmonitor", microservice="occnp_pcf_user", mktgVersion="1.0.0", namespace="s-euler", pod="test-occnp-udr-connector-868c766947-fr7zx", pod_template_hash="868c766947", queryParameters="service-names,group-id-list,target-nf-type,requester-nf-type", targetNfType="UDR", vendor="Oracle"}

Table 9-432 ocpm_nfDiscovery_response_total

Field	Details
Description	PCF measures the total number of successful/failed discovery query of UDR using "group-id-list".
Type	Counter
Dimensions	• queryParameters • responseCode • target_nf_type • cause
Example	ocpm_nfDiscovery_response_total{app_kubernetes_io_instance="test", app_kubernetes_io_managed_by="Helm", app_kubernetes_io_name="user-service", app_kubernetes_io_part_of="occnp", app_kubernetes_io_version="1.0.0", application="occnp", container="user-service", endpoint="cnc-metrics", engVersion="23.4.0-ocngf-incredibles", exported_application="pcf_userservice", helm_sh_chart="user-service-23.4.0-ocngf-incredibles", hostname="test-occnp-udr-connector-868c766947-fr7zx", instance="10.233.93.4:9000", io_kompose_service="test-occnp-udr-connector", job="occne-infra/occne-nf-cnc-podmonitor", microservice="occnp_pcf_user", mktgVersion="1.0.0", namespace="s-euler", pod="test-occnp-udr-connector-868c766947-fr7zx", pod_template_hash="868c766947", queryParameters="service-names,group-id-list,target-nf-type,requester-nf-type", responseCode="424", targetNfType="UDR", vendor="Oracle"}

Table 9-433 occnp_group_id_list_discovery_header_sent_total

Field	Details
Description	PCF measures the number of GET or POST request sent to UDR with 3gpp-Sbi-Discovery-group-Id-list for delegated discovery of UDR toward SCP.
Type	Counter
Dimensions	• operation_type • target_nf_type

Table 9-433 (Cont.) occnp_group_id_list_discovery_header_sent_total

Field	Details
Example	occnp_group_id_list_discovery_header_sent_total{app_kubernetes_io_instance="test", app_kubernetes_io_managed_by="Helm", app_kubernetes_io_name="user-service", app_kubernetes_io_part_of="occnp", app_kubernetes_io_version="1.0.0", application="occnp", container="user-service", endpoint="cnc-metrics", engVersion="23.4.0-ocngf-incredibles", exported_application="pcf_userservice", helm_sh_chart="user-service-23.4.0-ocngf-incredibles", hostname="test-occnp-udr-connector-79df49b796-gc77c", instance="10.233.72.203:9000", io_kompose_service="test-occnp-udr-connector", job="occne-infra/occne-nf-cnc-podmonitor", microservice="occnp_pcf_user", mktgVersion="1.0.0", namespace="s-euler", operation_type="get", pod="test-occnp-udr-connector-79df49b796-gc77c", pod_template_hash="79df49b796", vendor="Oracle"}

9.29 User-Agent Header Metrics

Table 9-434 oc.ingressgateway.http.requests

Field	Details
Description	This metric is pegged for every incoming request. If the User-Agent header is not present, then UNKNOWN will be pegged. This will be independent of the User Agent validation feature at the Ingress Gateway.
Type	Counter
Dimensions	• consumerNfType • consumerInstanceId • ConsumerFqdn

9.30 NWDAF Agent Metrics

The following table describes the NWDAF Agent metrics and their respective dimensions:

Table 9-435 subscription_failure

Field	Details
Description	Gauge of failures for configured slice load levels doing autonomous subscription during configuration changes.
Type	Gauge
Dimensions	• serviceResource • snssai • operationType • responseCode • requestType
Example	subscription_failure{operationType="POST",requestType="autonomous",responseCode="500",serviceResource="SLICE_LOAD_LEVEL",snssai="14-aaaa14",} 1.0

Table 9-436 http_backend_request_total

Field	Details
Description	Counter for backend requests
Type	Counter
Dimensions	• serviceNameNon3gpp • serviceResource • operationType
Example	http_backend_request_total{operationType="POST",serviceNameNon3gpp="autonomous",serviceResource="SLICE_LOAD_LEVEL",} 26.0

Table 9-437 http_backend_response_total

Field	Details
Description	Counter for responses to backend requests
Type	Counter
Dimensions	• serviceNameNon3gpp • serviceResource • serviceVersion • responseCode
Example	http_backend_response_total{operationType="POST",responseCode="201",serviceNameNon3gpp="autonomous",serviceResource="SLICE_LOAD_LEVEL",} 25.0

Table 9-438 http_in_conn_request_total

Field	Details
Description	Counter for NWDAF notification requests
Type	Counter
Dimensions	• serviceName3gpp • serviceResource • serviceVersion • operationType
Example	http_in_conn_request_total{operationType="POST",serviceName3gpp="NWDAF",serviceResource="NF_LOAD",serviceVersion="v1",} 1.0

Table 9-439 http_in_conn_response_total

Field	Details
Description	Counter for responses to NWDAF notification requests
Type	Counter
Dimensions	• serviceName3gpp • serviceResource • serviceVersion • operationType • responseCode

Table 9-439 (Cont.) http_in_conn_response_total

Field	Details
Example	http_in_conn_response_total{operationType="POST",responseCode="204",serviceName3gpp="NWDAF",serviceResource="NF_LOAD",serviceVersion="v1",} 1.0

Table 9-440 http_out_conn_request_total

Field	Details
Description	Counter for requests sent from nwdaf-agent to NWDAF
Type	Counter
Dimensions	• serviceName3gpp • serviceResource • serviceVersion • host • instanceId • isAlternateRoute • isRetryAttempt • operationType
Example	http_out_conn_request_total{host="floki-occnp-egress-gateway/nnwdaf-eventssubscription/v1",instanceId="fe7d992b-0541-4c7d-ab84-c6d70b1b0621",isAlternateRoute="",isRetryAttempt="",operationType="POST",serviceName3gpp="NWDAF",serviceResource="SLICE_LOAD_LEVEL",serviceVersion="v1",} 26.0

Table 9-441 http_out_conn_response_total

Field	Details
Description	Counter for responses from NWDAF
Type	Counter
Dimensions	• serviceName3gpp • serviceResource • serviceVersion • host • instanceId • isAlternateRoute • isRetryAttempt • operationType • responseCode
Example	http_out_conn_response_total{host="floki-occnp-egress-gateway/nnwdaf-eventssubscription/v1",instanceId="fe7d992b-0541-4c7d-ab84-c6d70b1b0621",isAlternateRoute="",isRetryAttempt="",operationType="POST",responseCode="500",serviceName3gpp="NWDAF",serviceResource="SLICE_LOAD_LEVEL",serviceVersion="v1",} 1.0

Table 9-442 topic_version

Field	Details
Description	Describes the current applied version of a given topic (mentioned in dimension topic_name) into the pod.
Type	Gauge
Dimensions	• Service Name • Pod Name

9.31 PRE Metrics

The following table describes the PRE Service metrics and respective dimensions:

Table 9-443 http_in_conn_request_total

Field	Details
Description	The requests received by a Microservice from the downstream application.
Type	Counter
Dimensions	• operationType • servicenameNon3gpp • workerId
Example	http_in_conn_request_total{operationType="post",servicenameNon3gpp="pcf-sm",workerId="1"}1

Table 9-444 http_in_conn_response_total

Field	Details
Description	The response sent for incoming requests to a micro-service.
Type	Counter
Dimensions	• operationType • servicenameNon3gpp • responseCode • workerId
Example	http_in_conn_response_total{operationType="post",servicenameNon3gpp="pcf-sm",responseCode="200",workerId="1"}1

Table 9-445 http_in_conn_processing_latency_ms

Field	Details
Description	The overall response time for request, received from the downstream application.
Type	Histogram
Dimensions	• le • operationType • servicenameNon3gpp • workerId

Table 9-445 (Cont.) http_in_conn_processing_latency_ms

Field	Details
Examples	• http_in_conn_processing_latency_ms_bucket{le="2",operationType="post",servicenameNon3gpp="pcf-sm",workerId="1"}0 • http_in_conn_processing_latency_ms_bucket{le="5",operationType="post",servicenameNon3gpp="pcf-sm",workerId="1"}1 • http_in_conn_processing_latency_ms_bucket{le="10",operationType="post",servicenameNon3gpp="pcf-sm",workerId="1"}1 • http_in_conn_processing_latency_ms_bucket{le="20",operationType="post",servicenameNon3gpp="pcf-sm",workerId="1"}3 • http_in_conn_processing_latency_ms_bucket{le="50",operationType="post",servicenameNon3gpp="pcf-sm",workerId="1"}5 • http_in_conn_processing_latency_ms_bucket{le="100",operationType="post",servicenameNon3gpp="pcf-sm",workerId="1"}5 • http_in_conn_processing_latency_ms_bucket{le="200",operationType="post",servicenameNon3gpp="pcf-sm",workerId="1"}5 • http_in_conn_processing_latency_ms_bucket{le="500",operationType="post",servicenameNon3gpp="pcf-sm",workerId="1"}5 • http_in_conn_processing_latency_ms_bucket{le="1000",operationType="post",servicenameNon3gpp="pcf-sm",workerId="1"}5 • http_in_conn_processing_latency_ms_bucket{le="2000",operationType="post",servicenameNon3gpp="pcf-sm",workerId="1"}5 • http_in_conn_processing_latency_ms_bucket{le="+Inf",operationType="post",servicenameNon3gpp="pcf-sm",workerId="1"}5 • http_in_conn_processing_latency_ms_sum{operationType="post",servicenameNon3gpp="pcf-sm",workerId="1"}82 • http_in_conn_processing_latency_ms_count{operationType="post",servicenameNon3gpp="pcf-sm",workerId="1"}5

Table 9-446 http_out_conn_request_total

Field	Details
Description	The requests sent by a micro-service to upstream.
Type	Counter
Dimensions	• operationType • host • httpVersion • isRetryAttempt • servicenameNon3gpp • workerId
	http_out_conn_request_total{operationType="POST",host="localhost:8080",httpVersion="1",isRetryAttempt="false",servicenameNon3gpp="PER",workerId="1"}1

Table 9-447 http_out_conn_response_total

Field	Details
Description	The response received for upstream requests to a micro-service.
Type	Counter

Table 9-447 (Cont.) http_out_conn_response_total

Field	Details
Dimensions	• operationType • responseCode • host • httpVersion • isRetryAttempt • servicenameNon3gpp • workerId
Example	http_out_conn_response_total{operationType="POST",responseCode="502",host="localhost:8080",httpVersion="1",isRetryAttempt="false",servicenameNon3gpp="PER",workerId="1"}1

Table 9-448 http_out_conn_processing_latency_ms

Field	Details
Description	The overall response time for request towards upstream.
Type	Histogram
Dimensions	• le • operationType • responseCode • host • httpVersion • isRetryAttempt • servicenameNon3gpp • workerId

Table 9-448 (Cont.) http_out_conn_processing_latency_ms

Field	Details
Examples	• http_out_conn_processing_latency_ms_bucket{le="2",operationType="POST",responseCode="502",host="localhost:8080",httpVersion="1",isRetryAttempt="false",servicenameNon3gpp="PER",workerId="1"}0 • http_out_conn_processing_latency_ms_bucket{le="5",operationType="POST",responseCode="502",host="localhost:8080",httpVersion="1",isRetryAttempt="false",servicenameNon3gpp="PER",workerId="1"}2 • http_out_conn_processing_latency_ms_bucket{le="10",operationType="POST",responseCode="502",host="localhost:8080",httpVersion="1",isRetryAttempt="false",servicenameNon3gpp="PER",workerId="1"}3 • http_out_conn_processing_latency_ms_bucket{le="20",operationType="POST",responseCode="502",host="localhost:8080",httpVersion="1",isRetryAttempt="false",servicenameNon3gpp="PER",workerId="1"}5 • http_out_conn_processing_latency_ms_bucket{le="50",operationType="POST",responseCode="502",host="localhost:8080",httpVersion="1",isRetryAttempt="false",servicenameNon3gpp="PER",workerId="1"}5 • http_out_conn_processing_latency_ms_bucket{le="100",operationType="POST",responseCode="502",host="localhost:8080",httpVersion="1",isRetryAttempt="false",servicenameNon3gpp="PER",workerId="1"}5 • http_out_conn_processing_latency_ms_bucket{le="200",operationType="POST",responseCode="502",host="localhost:8080",httpVersion="1",isRetryAttempt="false",servicenameNon3gpp="PER",workerId="1"}5 • http_out_conn_processing_latency_ms_bucket{le="500",operationType="POST",responseCode="502",host="localhost:8080",httpVersion="1",isRetryAttempt="false",servicenameNon3gpp="PER",workerId="1"}5 • http_out_conn_processing_latency_ms_bucket{le="1000",operationType="POST",responseCode="502",host="localhost:8080",httpVersion="1",isRetryAttempt="false",servicenameNon3gpp="PER",workerId="1"}5 • http_out_conn_processing_latency_ms_bucket{le="2000",operationType="POST",responseCode="502",host="localhost:8080",httpVersion="1",isRetryAttempt="false",servicenameNon3gpp="PER",workerId="1"}5 • http_out_conn_processing_latency_ms_bucket{le="+Inf",operationType="POST",responseCode="502",host="localhost:8080",httpVersion="1",isRetryAttempt="false",servicenameNon3gpp="PER",workerId="1"} 5 • http_out_conn_processing_latency_ms_sum{operationType="POST",responseCode="502",host="localhost:8080",httpVersion="1",isRetryAttempt="false",servicenameNon3gpp="PER",workerId="1"}41 • http_out_conn_processing_latency_ms_count{operationType="POST",responseCode="502",host="localhost:8080",httpVersion="1",isRetryAttempt="false",servicenameNon3gpp="PER",workerId="1"}5

Table 9-449 occnp_policy_processing_latency_ms

Field	Details
Description	Overall response time for request received from downstream when executing Policy. **Note**: This metric is associated with 'Call Policy' block. For more information, see "Public Category" section in Oracle Communications Cloud Native Core, Converged Policy Design Guide.
Type	Histogram

Table 9-449 (Cont.) occnp_policy_processing_latency_ms

Field	Details
Dimensions	• le • operationType • policyName • servicenameNon3gpp • workerId
Examples	• occnp_policy_processing_latency_ms_bucket{le="2", operationType="post", policyName="AM_Policy_Audit", servicenameNon3gpp="pcf-am", workerId="2"}0 • occnp_policy_processing_latency_ms_bucket{le="5", operationType="post", policyName="AM_Policy_Audit", servicenameNon3gpp="pcf-am", workerId="2"}1 • occnp_policy_processing_latency_ms_bucket{le="10", operationType="post", policyName="AM_Policy_Audit", servicenameNon3gpp="pcf-am", workerId="2"}1 • occnp_policy_processing_latency_ms_bucket{le="20", operationType="post", policyName="AM_Policy_Audit", servicenameNon3gpp="pcf-am", workerId="2"}1 • occnp_policy_processing_latency_ms_bucket{le="50", operationType="post", policyName="AM_Policy_Audit", servicenameNon3gpp="pcf-am", workerId="2"}1 • occnp_policy_processing_latency_ms_bucket{le="100", operationType="post", policyName="AM_Policy_Audit", servicenameNon3gpp="pcf-am", workerId="2"}1 • occnp_policy_processing_latency_ms_bucket{le="200", operationType="post", policyName="AM_Policy_Audit", servicenameNon3gpp="pcf-am", workerId="2"}1 • occnp_policy_processing_latency_ms_bucket{le="500", operationType="post", policyName="AM_Policy_Audit", servicenameNon3gpp="pcf-am", workerId="2"}1 • occnp_policy_processing_latency_ms_bucket{le="1000", operationType="post", policyName="AM_Policy_Audit", servicenameNon3gpp="pcf-am", workerId="2"}1 • occnp_policy_processing_latency_ms_bucket{le="2000", operationType="post", policyName="AM_Policy_Audit", servicenameNon3gpp="pcf-am", workerId="2"}1 • occnp_policy_processing_latency_ms_bucket{le="+Inf", operationType="post", policyName="AM_Policy_Audit", servicenameNon3gpp="pcf-am", workerId="2"}1 • occnp_policy_processing_latency_ms_sum{operationType="post", policyName="AM_Policy_Audit", servicenameNon3gpp="pcf-am", workerId="2"}1 • occnp_policy_processing_latency_ms_count{ operationType="post", policyName="AM_Policy_Audit", servicenameNon3gpp="pcf-am", workerId="2"}1 **Note**: This metric is associated with 'Call Policy' block. For more information, see "Public Category" section in Oracle Communications Cloud Native Core, Converged Policy Design Guide.

Table 9-450 occnp_block_counter_label

Field	Details
Description	Tracks total number of given custom block is invoked for corresponding label. **Note**: This metric is associated with 'Increment Counter Label' block. For more information, see "Public Category" section in Oracle Communications Cloud Native Core, Converged Policy Design Guide.
Type	Counter
Dimensions	• serviceType • operationType • workerId • label • subLabel
Example	occnp_block_counter_label{serviceType="pcf-sm",operationType="post",workerId="2",label="lml",subLabel=""}1

Table 9-451 occnp_block_exec_time_ns

Field	Details
Description	Tracks overall time taken by all the blocks captured as part of this block. **Note**: This metric is associated with 'Time' block. For more information, see "Public Category" section in Oracle Communications Cloud Native Core, Converged Policy Design Guide.
Type	Histogram
Dimensions	• serviceType • operationType • workerId • label • subLabel

Table 9-451 (Cont.) occnp_block_exec_time_ns

Field	Details
Example	• occnp_block_exec_time_ns_bucket{le="2",serviceType="pcf-sm",operationType="post",workerId="2",label="lnl",subLabel=""}1 • occnp_block_exec_time_ns_bucket{le="5",serviceType="pcf-sm",operationType="post",workerId="2",label="lnl",subLabel=""}1 • occnp_block_exec_time_ns_bucket{le="10",serviceType="pcf-sm",operationType="post",workerId="2",label="lnl",subLabel=""}1 • occnp_block_exec_time_ns_bucket{le="20",serviceType="pcf-sm",operationType="post",workerId="2",label="lnl",subLabel=""}1 • occnp_block_exec_time_ns_bucket{le="50",serviceType="pcf-sm",operationType="post",workerId="2",label="lnl",subLabel=""}1 • occnp_block_exec_time_ns_bucket{le="100",serviceType="pcf-sm",operationType="post",workerId="2",label="lnl",subLabel=""}1 • occnp_block_exec_time_ns_bucket{le="200",serviceType="pcf-sm",operationType="post",workerId="2",label="lnl",subLabel=""}1 • occnp_block_exec_time_ns_bucket{le="500",serviceType="pcf-sm",operationType="post",workerId="2",label="lnl",subLabel=""}1 • occnp_block_exec_time_ns_bucket{le="1000",serviceType="pcf-sm",operationType="post",workerId="2",label="lnl",subLabel=""}1 • occnp_block_exec_time_ns_bucket{le="2000",serviceType="pcf-sm",operationType="post",workerId="2",label="lnl",subLabel=""}1 • occnp_block_exec_time_ns_bucket{le="+Inf",serviceType="pcf-sm",operationType="post",workerId="2",label="lnl",subLabel=""}1 • occnp_block_exec_time_ns_sum{serviceType="pcf-sm",operationType="post",workerId="2",label="lnl",subLabel=""}1 • occnp_block_exec_time_ns_count{serviceType="pcf-sm",operationType="post",workerId="2",label="lnl",subLabel=""}1

For more information about Dimensions, see CNC Policy Metrics.

9.32 NRF Client Metrics

NF status and NF load metrics:

Table 9-452 nrfclient_perf_info_nf_profile_load

Field	Details
Description	The current Load of the NF.
Type	Gauge
Dimensions	

Table 9-453 nrfclient_current_nf_status

Field	Details
Description	The current operative status of the NF. The gauge shall be indicate the status as below: • 0 - REGISTERED • 1 - DEREGISTERED • 2 - SUSPENDED • 3 - UNDISCOVERABLE • 4 - UNKNOWN
Type	Gauge
Dimensions	• NfType - The NF's NfType as present in the registered NfProfile. • NfInstanceID - The NF's nfInstanceId as present in the registered NfProfile. • NfFqdn - The NF's FQDN as present in the registered NfProfile.

Table 9-454 nrfclient_nf_status_with_nrf

Field	Details
Description	The operative status of the NF communicated to the NRF. The gauge shall be indicate the status as below: • 0 - REGISTERED • 1 - DEREGISTERED • 2 - SUSPENDED • 3 - UNDISCOVERABLE • 4 - UNKNOWN
Type	Gauge
Dimensions	• NfType - The NF's NfType as present in the registered NfProfile. • NfInstanceID - The NF's nfInstanceId as present in the registered NfProfile. • NfFqdn - The NF's FQDN as present in the registered NfProfile.

NRF Health Status:

Table 9-455 nrfclient_nrf_operative_status

Field	Details
Description	The current operative status of the NRF Instance. If the metric has value • 0 - NRF is unavailable or unhealthy • 1 - NRF is available or healthy
Type	Gauge
Dimensions	NrfUri - URI of the NRF Instance

Table 9-456 nrfclient_nrf_status_total

Field	Details
Description	Total number of times an NRF instance is marked as healthy or unhealthy. The apiRoot is specified in the format *'scheme'://'fqdn':'port'*. **Note**: If health check procedure is disabled, all NRF instances are marked as HEALTHY after successful NF registration.
Type	Counter

Table 9-456 (Cont.) nrfclient_nrf_status_total

Field	Details
Dimensions	NrfUri- URI of the NRF InstanceHealthStatus FailureReason - Reason for the status

Table 9-457 nrfclient_nrf_successive_healthy_count

Field	Details
Description	The metric shows the consecutive number of times the NRF is considered as healthy. The metric has a minimum value of 0 and maximum value of *healthCheckCount*.
Type	Counter
Dimensions	NrfUri- URI of the NRF Instance

Table 9-458 nrfclient_nrf_successive_unhealthy_count

Field	Details
Description	The metric shows the consecutive number of times the NRF is considered as unhealthy
Type	Counter
Dimensions	NrfUri- URI of the NRF Instance

NF - NRF-Client metrics:

Table 9-459 nrfclient_on_demand_conn_in_request_total

Field	Details
Description	Total number of on-demand requests received from the backend NF to NRF Client.
Type	Counter
Dimensions	• MessageType - The service request Type. • NfType - The NF's NfType as present in the registered NfProfile. • NfInstanceID - The NF's nfInstanceId as present in the registered NfProfile. • NfFqdn - The NF's FQDN as present in the registered NfProfile.

Table 9-460 nrfclient_on_demand_conn_out_response_total

Field	Details
Description	Total number of on-demand responses sent to the backend NF to NRF Client.
Type	Counter
Dimensions	• MessageType - The service request Type. • NfType - The NF's NfType as present in the registered NfProfile. • NfInstanceID - The NF's nfInstanceId as present in the registered NfProfile. • NfFqdn - The NF's FQDN as present in the registered NfProfile. • StatusCode - The HttpStatusCode as received from the NRF or generated by NRF-client.

Table 9-461 nrfclient_on_demand_processing_latency_ms

Field	Details
Description	Total message processing time duration in milliseconds.
Type	Histogram
Dimensions	• MessageType - The service request Type. • NfType - The NF's NfType as present in the registered NfProfile. • NfInstanceID - The NF's nfInstanceId as present in the registered NfProfile. • NfFqdn - The NF's FQDN as present in the registered NfProfile.

Table 9-462 ocpm_nrf_tracing_request_timeout_total

Field	Details
Description	Total number of requests timeout sent to the backend NF from NRF Client.
Type	Counter
Dimensions	• MessageType - The service request Type. • NfType - The NF's NfType as present in the registered NfProfile. • NfInstanceID - The NF's nfInstanceId as present in the registered NfProfile. • NfFqdn - The NF's FQDN as present in the registered NfProfile.

NRF-Client - NRF metrics:

Table 9-463 nrfclient_nw_conn_out_request_total

Field	Details
Description	Total number of times NRF-client has sent a request to NRF. This includes autonomous requests as well as on-demand requests.
Type	Counter
Dimensions	• MessageType - The service request Type. • NfType - The NF's NfType as present in the registered NfProfile. • NfInstanceID - The NF's nfInstanceId as present in the registered NfProfile. • NfFqdn - The NF's FQDN as present in the registered NfProfile.

Table 9-464 nrfclient_nw_conn_in_response_total

Field	Details
Description	Total number of times NRF-client has received a response from NRF.
Type	Counter
Dimensions	• MessageType - The service request Type. • NfType - The NF's NfType as present in the registered NfProfile. • NfInstanceID - The NF's nfInstanceId as present in the registered NfProfile. • NfFqdn - The NF's FQDN as present in the registered NfProfile. • StatusCode - The HttpStatusCode as received from the NRF or generated by NRF-client.
Example	occnp_nrfclient_nw_conn_in_response_total{ApplicationError="INVALID_API",MessageType="NfDiscovery",NfFqdn="occnp-ocpm-ingress-gateway.ocpcf.svc",NfInstanceID="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",NfType="PCF",StatusCode="400",} 2.0

Table 9-465 nrfclient_nw_conn_in_notify_request_total

Field	Details
Description	Total number of nfStatusNotify requests received from NRF.
Type	Counter
Dimensions	• EventType - The EventType for which the notification is triggered. • NfInstanceID - The NfInstanceId for which the notification is triggered.

Table 9-466 nrfclient_nw_conn_out_notify_response_total

Field	Details
Description	Total number of *nfStatusNotify* responses sent to NRF.
Type	Counter
Dimensions	• EventType - The EventType for which the notification is triggered. • NfInstanceID - The NfInstanceId for which the notification is triggered. • HttpStatusCode - The HttpStatusCode sent by NRF-Client.
Example	occnp_nrfclient_nw_conn_out_notify_response_total # TYPE occnp_nrfclient_nw_conn_out_notify_response_total counter occnp_nrfclient_nw_conn_out_notify_response_total{EventType="NF_REGISTERED",HttpStatusCode="400",NfInstanceID="4947a69a-f61b-4bc1-b9da-47c9c5d14b64",} 1.0

Table 9-467 nrfclient_network_message_processing_latency

Field	Details
Description	Total message processing time duration.
Type	Histogram
Dimensions	• MessageType - The service request Type. • NfType - The NF's NfType as present in the registered NfProfile. • NfInstanceID - The NF's nfInstanceId as present in the registered NfProfile. • NfFqdn - The NF's FQDN as present in the registered NfProfile.

Table 9-468 occnp_nrfclient_discovery_cache_support_force_discovery_total

Field	Details
Description	Indicates the total number of requests received from backend services with header OC-Force-Rediscovery is enabled, and response is retrieved from NRF.
Type	Counter
Dimensions	• NfType • NfInstanceID • NfFqdn • QueryParams

Table 9-469 occnp_nrfclient_discovery_cache_support_cache_hit_total

Field	Details
Description	Indicates the total number of requests for which discovery response is returned from the cache.
Type	Counter
Dimensions	• NfType • NfInstanceID • NfFqdn • QueryParams

Table 9-470 occnp_nrfclient_discovery_cache_support_about_to_expire_total

Field	Details
Description	Indicates the total number of cache records identified as about to be expired. This metric counts cases when a cached discovery response is eligible to be returned but also triggers a background request to get an updated response.
Type	Counter
Dimensions	• NfType • NfInstanceID • NfFqdn • QueryParams

Table 9-471 occnp_nrfclient_discovery_cache_support_expired_total

Field	Details
Description	Metric for expired cache record scenario. This metric counts cases when a cached discovery response is not eligible to be returned due to expiration and a new response must be retrieved again from NRF.
Type	Counter
Dimensions	• NfType • NfInstanceID • NfFqdn • QueryParams

Table 9-472 occnp_nrfclient.discovery.cache.support.failover

Field	Details
Description	Metric for cache failover scenario. This metric counts cases when the response was returned by the cache failover logic.
Type	Counter
Dimensions	• NfType • NfInstanceID • NfFqdn • QueryParams

Table 9-473 occnp_nrfclient_discovery_cache_support_cache_non_cache_total

Field	Details
Description	Metric for non-cache scenario. This metrics counts cases when the discovery request parameters do not fully match with the configurated parameters.
Type	Counter
Dimensions	• NfType • NfInstanceID • NfFqdn

Table 9-474 occnp_nrfclient_discovery_cache_support_empty_response_total

Field	Details
Description	Metric for empty response scenario. This metric will count cases when an empty response was returned to NF from Cache or NRF.
Type	Counter
Dimensions	• NfType • NfInstanceID • NfFqdn • QueryParams • TypeSource

Table 9-475 occnp_nrfclient_discovery_cache_support_cache_lookup_seconds

Field	Details
Description	Metric for cache lookups. This metric measures time taken to perform cache lookups from cache or DB without considering the complete request flow.
Type	Histogram
Dimensions	• NfType • NfInstanceID • NfFqdn

Table 9-476 occnp_nrfclient_discovery_cache_support_backend_response_seconds

Field	Details
Description	Metric for cached and non-cached responses returned to backend NF. This metric measures the time taken to perform the complete request flow when response was taken from cache or DB or NRF.
Type	Histogram
Dimensions	• NfType • NfInstanceID • NfFqdn • TypeSource

Table 9-477 occnp_nrfclient_discovery_cache_support_cache_lookup_seconds_bucket

Field	Details
Description	This metric is used to perform the cache lookup.
Type	Histogram
Dimensions	• NfType • NfInstanceID • NfFqdn
Examples	occnp_nrfclient_discovery_cache_support_cache_lookup_seconds_bucket{NfFqdn="occnp-ocpm-ingress-gateway.ocpcf.svc",NfInstanceID="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",NfType="PCF",le="0.01",} 19.0

Table 9-478 occnp_nrfclient_discovery_cache_support_cache_lookup_seconds_count

Field	Details
Description	This metric is used to get a count of cache lookup requests.
Type	Histogram
Dimensions	• NfType • NfInstanceID • NfFqdn
Examples	occnp_nrfclient_discovery_cache_support_cache_lookup_seconds_count{NfFqdn="occnp-ocpm-ingress-gateway.ocpcf.svc",NfInstanceID="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",NfType="PCF",} 19.0

Table 9-479 occnp_nrfclient_discovery_cache_support_cache_lookup_seconds_max

Field	Details
Description	This metric provides the maximum value of cache lookup.
Type	Gauge
Dimensions	• NfType • NfInstanceID • NfFqdn
Examples	occnp_nrfclient_discovery_cache_support_cache_lookup_seconds_max{NfFqdn="occnp-ocpm-ingress-gateway.ocpcf.svc",NfInstanceID="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",NfType="PCF",} 1.7408E-5

Table 9-480 occnp_nrfclient_discovery_cache_support_cache_lookup_seconds_sum

Field	Details
Description	This metric provides the sum of cache lookup.
Type	Histogram
Dimensions	• NfType • NfInstanceID • NfFqdn

Table 9-480 (Cont.) occnp_nrfclient_discovery_cache_support_cache_lookup_seconds_sum

Field	Details
Examples	occnp_nrfclient_discovery_cache_support_cache_lookup_seconds_sum{NfFqdn="occnp-ocpm-ingress-gateway.ocpcf.svc",NfInstanceID="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",NfType="PCF",} 0.001040425

Table 9-481 occnp_nrfclient_discovery_cache_support_backend_response_seconds_bucket

Field	Details
Description	This metric is used to peg the backend response time for cached and non-cached responses
Type	Histogram
Dimensions	• NfType • NfInstanceID • NfFqdn • SourceType
Examples	occnp_nrfclient_discovery_cache_support_backend_response_seconds_bucket{NfFqdn="occnp-ocpm-ingress-gateway.ocpcf.svc",NfInstanceID="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",NfType="PCF",SourceType="Cache",le="0.01",} 19.0

Table 9-482 occnp_nrfclient_discovery_cache_support_backend_response_seconds_count

Field	Details
Description	This metric provides the count of cached responses.
Type	Histogram
Dimensions	• NfType • NfInstanceID • NfFqdn • SourceType
Examples	occnp_nrfclient_discovery_cache_support_backend_response_seconds_count{NfFqdn="occnp-ocpm-ingress-gateway.ocpcf.svc",NfInstanceID="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",NfType="PCF",SourceType="Cache",} 19.0

Table 9-483 occnp_nrfclient_discovery_cache_support_backend_response_seconds_max

Field	Details
Description	This metric provides maximum backend response time for cached responses.
Type	Gauge
Dimensions	• NfType • NfInstanceID • NfFqdn • SourceType

Table 9-483 (Cont.) occnp_nrfclient_discovery_cache_support_backend_response_seconds_max

Field	Details
Examples	occnp_nrfclient_discovery_cache_support_backend_response_seconds_max{NfFqdn="occnp-ocpm-ingress-gateway.ocpcf.svc",NfInstanceID="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",NfType="PCF",SourceType="Cache",} 3.2963E-5

Table 9-484 occnp_nrfclient_discovery_cache_support_backend_response_seconds_sum

Field	Details
Description	This metric provides the sum of backend response time for cached responses.
Type	Histogram
Dimensions	• NfType • NfInstanceID • NfFqdn • SourceType
Examples	occnp_nrfclient_discovery_cache_support_backend_response_seconds_sum{NfFqdn="occnp-ocpm-ingress-gateway.ocpcf.svc",NfInstanceID="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",NfType="PCF",SourceType="Cache",} 0.001164615

Table 9-485 occnp_nrfclient_discovery_cache_support_profiles_per_response_bucket

Field	Details
Description	This metric is used to measure the number of producer profiles returned to backend NF.
Type	Histogram
Dimensions	• NfType • NfInstanceID • NfFqdn • SourceType
Examples	occnp_nrfclient_discovery_cache_support_profiles_per_response_bucket{NfFqdn="occnp-ocpm-ingress-gateway.ocpcf.svc",NfInstanceID="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",NfType="PCF",SourceType="Cache",le="1.0",} 1.0

Table 9-486 occnp_nrfclient_discovery_cache_support_profiles_per_response_count

Field	Details
Description	This metric provides the count of number of producer profiles returned to backend NF for all responses.
Type	Histogram
Dimensions	• NfType • NfInstanceID • NfFqdn • SourceType

Table 9-486 (Cont.) occnp_nrfclient_discovery_cache_support_profiles_per_response_count

Field	Details
Examples	occnp_nrfclient_discovery_cache_support_profiles_per_response_count{NfFqdn="occnp-ocpm-ingress-gateway.ocpcf.svc",NfInstanceID="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",NfType="PCF",SourceType="Cache",} 19.0

Table 9-487 occnp_nrfclient_discovery_cache_support_profiles_per_response_max

Field	Details
Description	This metric provides the maximum number of producer profiles returned to backend NF in particular response.
Type	Gauge
Dimensions	• NfType • NfInstanceID • NfFqdn • SourceType
Examples	occnp_nrfclient_discovery_cache_support_profiles_per_response_max{NfFqdn="occnp-ocpm-ingress-gateway.ocpcf.svc",NfInstanceID="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",NfType="PCF",SourceType="Cache",} 5.0

Table 9-488 occnp_nrfclient_discovery_cache_support_profiles_per_response_sum

Field	Details
Description	This metric provides the sum of producer profiles returned to backend NF in all responses.
Type	Histogram
Dimensions	• NfType • NfInstanceID • NfFqdn • SourceType
Examples	occnp_nrfclient_discovery_cache_support_profiles_per_response_sum{NfFqdn="occnp-ocpm-ingress-gateway.ocpcf.svc",NfInstanceID="fe7d992b-0541-4c7d-ab84-c6d70b1b0123",NfType="PCF",SourceType="NRF",} 420.0

Table 9-489 occnp_nrfclient_nfUpdate_status

Field	Details
Description	This metric marks the status of the NfUpdate Service Operation.
Type	Gauge

Table 9-489 (Cont.) occnp_nrfclient_nfUpdate_status

Field	Details
Dimensions	• NfFqdn • NfInstanceID • NfType • app_kubernetes_io_instance • app_kubernetes_io_managed_by • app_kubernetes_io_name • app_kubernetes_io_part_of • app_kubernetes_io_version • application • container • endpoint • engVersion • helm_sh_chart • instance • io_kompose_service • job • microservice • mktgVersion • namespace • pod • pod_template_hash • vendor
Examples	occnp_nrfclient_nfUpdate_status{NfFqdn="occnp-occnp-ingress-gateway.occnp", NfInstanceID="fe7d992b-0541-4c7d-ab84-c6d2410a2410", NfType="PCF", app_kubernetes_io_instance="occnp", app_kubernetes_io_managed_by="Helm", app_kubernetes_io_name="nrf-client-nfmanagement", app_kubernetes_io_part_of="nrf-client", app_kubernetes_io_version="24.1.5.0.0", application="occnp", container="nrf-client-nfmanagement", endpoint="cnc-metrics", engVersion="24.1.5", helm_sh_chart="nrf-client-nfmanagement-24.1.5", instance="10.233.109.152:9000", io_kompose_service="occnp-occnp-nrf-client-nfmanagement", job="occne-infra/occne-nf-cnc-podmonitor", microservice="occnp-nrf-client-nfmanagement", mktgVersion="24.1.5.0.0", namespace="occnp", pod="occnp-occnp-nrf-client-nfmanagement-8f94667cf-bbhls", pod_template_hash="8f94667cf", vendor="Oracle"}

9.33 Error Mapping Metrics

Table 9-490 error_handler_in_total

Field	Details
Description	Indicates the total number of client requests.
Type	Counter
Dimensions	• application • application_exception • wrapped_exception
Example	error_handler_in_total{application="policyds",application_exception="HttpException",wrapped_exception="TimeoutException",} 1.0

Table 9-491 error_handler_exec_total

Field	Details
Description	Metric on error handling performed by framework.
Type	Counter
Dimensions	• application • application_exception • error_type • operation • origin • rule_name • source_interface • target • wrapped_exception
Example	error_handler_exec_total{application="policyds",application_exception="HttpException",error_type="INTERNAL",operation="LOOKUP",origin="HTTP",rule_name="HTTP_REQUEST",source_interface="POLICY",status="502",target_interface="BSF",wrapped_exception="BadGateway",} 1.0

Table 9-492 error_handler_out_total

Field	Details
Description	Metric on completion of error handling.
Type	Counter
Dimensions	• application • application_exception • error_resolved • wrapped_exception
Example	error_handler_out_total{application="policyds",application_exception="HttpException",error_resolved="true",wrapped_exception="BadRequest",} 1.0

9.34 Metrics for Automated Certificate Lifecycle Management

The following metrics are used to support automated certificate lifecycle management for Policy:

Table 9-493 oc_certificatemanagement_tls_certificate_info

Field	Details
Description	This metric is used to peg status of TLS certificates.
Type	Guage
Dimensions	• CertificateName • SecretName • Status (VALID,CORRUPT,MISSING,EXPIRED) • Service (IngressGateway,EgressGateway)
Example	

10 CNC Policy KPIs

This section provides information about Key Performance Indicators (KPIs) used for Cloud Native Core Policy.

> **Note**
>
> Sample CNC Policy dashboard for Grafana is delivered to the customer through CNC Policy Custom Templates. The metrics and functions used to achieve KPIs are covered in CNC Policy Custom Templates as well.

CPU and Memory Usage

KPI Name	KPI Details	Metric used for KPI
POD-Count	Measures the number of PCF SM Service pods available in the system.	count(container_memory_usage_bytes{container_name='pcf-smservice',namespace=\"$namespace\"})
POD-Count	Measures the number of PCF User Service pods available in the system.	count(container_memory_usage_bytes{container_name='pcf-userservice',namespace=\"$namespace\"})
POD-Count	Measures the number of PRE pods available in the system.	count(container_memory_usage_bytes{container_name='ocpm-pre',namespace=\"$namespace\"})
POD-Count	Measures the number of PCF AM Service pods available in the system.	count(container_memory_usage_bytes{container_name='pcf-amservice',namespace=\"$namespace\"})
POD-Count	Measures the number of NRF Client-NRF Discovery pods available in the system.	count(container_memory_usage_bytes{container_name='nrf-client-nfdiscovery',namespace=\"$namespace\"})
POD-Count	Measures the number of Ingress Gateway pods available in the system.	count(container_memory_usage_bytes{container_name='ingress-gateway',namespace=\"$namespace\"})
POD-Count	Measures the number of Egress Gateway pods available in the system.	count(container_memory_usage_bytes{container_name='egress-gateway',namespace=\"$namespace\"})

KPI Name	KPI Details	Metric used for KPI
Total TPS	Measures the rate of (Ingress + Egress + Diameter) Gateway requests received at CNC Policy.	sum(rate(occnp_oc_ingressgateway_http_requests_total{$namespacelbl="$namespace"}[2m])) + sum(rate(occnp_oc_egressgateway_http_requests_total{$namespacelbl="$namespace",Direction="egressOut"}[2m])) + sum(rate(occnp_diam_request_network_total{$namespacelbl="$namespace", appId!="0"}[2m]))
Memory-Usage	Measures the current memory usage in bytes.	sum(container_memory_usage_bytes{image!="",namespace=\"$namespace\"})
Memory-Usage	Measures the memory usage (in bytes) for the top 16 memory users by each pod.	topk(16,sum(container_memory_usage_bytes{image!="",namespace=\"$namespace\"}) by (container_name))
CPU-Usage	Measures the number of cores being used by each pod.	sum(rate(container_cpu_usage_seconds_total{image!="",namespace=\"$namespace\",container_name!='POD'}[2m])) by (container_name)

Session Management Service

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
SM Create Requests	Measures the rate of requests received at PCF to create SM policy association.	sum(rate(ocpm_ingress_request_total{kubernetes_namespace=\"$namespace\",servicename_3gpp=\"npcf-smpolicycontrol\",operation_type=\"create\",dnn=\"dnn1\",snssai=\"11-abc123\"}[2m]))	Create	Not Applicable

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
SM Create Success Response	Measures the rate of requests for which SM policy associations are created successfully.	sum (rate(ocpm_ingress_response_total{kubernetes_namespace=\"$namespace\",servicename_3gpp=\"npcf-smpolicycontrol\",operation_type=\"create\",response_code=\"2xx\"}[2m])) **Note**: [2m] determines the rate interval at which the value needs to be calculated. This value is strictly used for calculation purpose and can be modified accordingly as per customer requirements. It can be changed either before or after adding the sample to Policy Grafana dashboard.	Create	2xx
SM Create Failure Response	Measures the rate at which create requests for SM policy association are rejected by PCF due to certain errors.	sum (rate(ocpm_ingress_response_total{kubernetes_namespace=\"$namespace\",servicename_3gpp=\"npcf-smpolicycontrol\",operation_type=\"create\",response_code!=\"2xx\"}[2m]))	Create	2xx
SM Update Request	Measures the rate at which requests are received at PCF to update the SM Policy association data.	sum (rate(ocpm_ingress_request_total{kubernetes_namespace=\"$namespace\",servicename_3gpp=\"npcf-smpolicycontrol\",operation_type=\"update\",dnn=\"dnn1\",snssai=\"11-abc123\"}[2m]))	Update	Not Applicable

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
SM Update Success Response	Measures the rate at which requests for updating SM policy association (already existing) are updated successfully.	sum (rate(ocpm_ingress_response_total{kubernetes_namespace=\"$namespace\",servicename_3gpp=\"npcf-smpolicycontrol\",operation_type=\"update\",response_code=\"2xx\"}[2m]))	Update	2xx
SM Update Failure Response	Measures the rate at which requests for updating SM policy association (already existing) are rejected by PCF due to certain errors at the consumer end.	sum (rate(ocpm_ingress_response_total{kubernetes_namespace=\"$namespace\",servicename_3gpp=\"npcf-smpolicycontrol\",operation_type=\"update\",response_code=\"4xx\"}[2m]))	Update	4xx
SM Delete Requests	Measures the rate of requests received at PCF to delete (or deregister) SM policy association.	sum (rate(ocpm_ingress_request_total{kubernetes_namespace=\"$namespace\",servicename_3gpp=\"npcf-smpolicycontrol\",operation_type=\"delete\",dnn=\"dnn1\",snssai=\"11-abc123\"}[2m]))	Delete	Not Applicable
SM Delete Success Responses	Measures the rate at which requests for deleting SM policy association (already existing) are deleted successfully.	sum (rate(ocpm_ingress_response_total{kubernetes_namespace=\"$namespace\",servicename_3gpp=\"npcf-smpolicycontrol\",operation_type=\"delete\",response_code=\"2xx\"}[2m]))	Delete	2xx
SM Delete Failure Responses	Measures the rate at which requests for deleting SM policy association (already existing) are rejected by PCF due to certain errors at the consumer end.	sum (rate(ocpm_ingress_response_total{kubernetes_namespace=\"$namespace\",servicename_3gpp=\"npcf-smpolicycontrol\",operation_type=\"delete\",response_code!=\"2xx\"}[2m]))	Delete	2xx

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
SM Update Notify Requests	Measures the rate at which requests are initiated by PCF towards SMF to update the PCC rules PDU session related policy context.	sum (rate(ocpm_egress_request_total{kubernetes_namespace=\"$namespace\",servicename_3gpp=\"npcf-smpolicycontrol\",operation_type=\"update_notify\"}[2m]))	Update Notify	Not Applicable
SM Update Notify Success Response	Measures the rate at which update requests received at SMF (initiated by PCF) are updated successfully.	sum (rate(ocpm_egress_response_total{kubernetes_namespace=\"$namespace\",servicename_3gpp=\"npcf-smpolicycontrol\",operation_type=\"update_notify\",response_code=\"2xx\"}[2m]))	Update Notify	2xx
SM Update Notify Failure Response	Measures the rate at which update requests received at SMF (initiated by PCF) are rejected due to certain errors.	sum (rate(ocpm_egress_response_total{kubernetes_namespace=\"$namespace\",servicename_3gpp=\"npcf-smpolicycontrol\",operation_type=\"update_notify\",response_code!=\"2xx\"}[2m]))	Update Notify	2xx
SM Terminate Notify Requests	Measures the rate at which requests are initiated by PCF towards SMF to delete SM Policy association of a PDU session.	sum (rate(ocpm_egress_request_total{kubernetes_namespace=\"$namespace\",servicename_3gpp=\"npcf-smpolicycontrol\",operation_type=\"terminate_notify\"}[2m]))	Terminate Notify	Not Applicable
SM Terminate Notify Success Response	Measures the rate at which delete requests received at SMF (initiated by PCF) are processed successfully.	sum (rate(ocpm_egress_response_total{kubernetes_namespace=\"$namespace\",servicename_3gpp=\"npcf-smpolicycontrol\",operation_type=\"terminate_notify\",response_code=\"2xx\"}[2m]))	Terminate Notify	2xx

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
SM Terminate Notify Failure Response	Measures the rate at which delete requests received at SMF (initiated by PCF) are rejected due to certain errors.	sum (rate(ocpm_egress_response_total{kubernetes_namespace=\"$namespace\",servicename_3gpp=\"npcf-smpolicycontrol\",operation_type=\"terminate_notify\",response_code!=\"2xx\"}[2m]))	Terminate Notify	2xx

Diameter Gateway Request and Response

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
Rx AAR Initial Request	Measures the rate of Rx AAR initial messages received by Diameter Gateway.	sum (rate(ocpm_ingress_request_total{kubernetes_namespace=\"$namespace\",operation_type=\"create\",servicename_3gpp=\"rx\"}[2m]))	Create	Not Applicable
Rx AAA Initial Response Success	Measures the rate at which Diameter Gateway processes Rx AAR messages successfully by sending Rx AAA as initial response.	sum (rate(ocpm_ingress_response_total{kubernetes_namespace=\"$namespace\",operation_type=\"create\",servicename_3gpp=\"rx\",response_code=\"2xxx\"}[2m]))	Create	2xxx
Rx AAR Update Request	Measures the rate of Rx AAR update requests received by Diameter Gateway.	sum (rate(ocpm_ingress_request_total{kubernetes_namespace=\"$namespace\",operation_type=\"update\",servicename_3gpp=\"rx\"}[2m]))	Update	Not Applicable
Rx AAR Update Response Success	Measures the rate at which Diameter Gateway processes Rx AAR update requests successfully.	sum (rate(ocpm_ingress_response_total{kubernetes_namespace=\"$namespace\",operation_type=\"update\",servicename_3gpp=\"rx\",response_code=\"2xxx\"}[2m]))	Update	2xxx

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
Rx STR Request	Measures the rate of the Session-Termination-Request (STR) messages received by Diameter Gateway.	sum (rate(ocpm_ingress_request_total{kubernetes_namespace=\"$namespace\",operation_type=\"terminate\",servicename_3gpp=\"rx\",response_code=\"2xxx\"}[2m]))	Terminate	2xxx
Rx STR Response Success	Measures the rate at which Diameter Gateway processes Rx STR messages successfully.	sum (rate(ocpm_ingress_response_total{kubernetes_namespace=\"$namespace\",operation_type=\"terminate\",servicename_3gpp=\"rx\",response_code=\"2xxx\"}[2m]))	Terminate	2xxx

UDR Tracking Request and Response

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
Query SM Data UDR Request	Measures the number of query requests sent by PCF to UDR.	sum (rate(ocpm_udr_tracking_request_total{kubernetes_namespace=\"$namespace\",operation_type=\"get\",service_version=\"v1\",servicename_3gpp=\"nudr-dr\"}[2m]))	GET	Not Applicable
Unsubscribe UDR Request	Measures the number of unsubscribe requests sent by PCF to UDR.	sum(rate(ocpm_udr_tracking_request_total{kubernetes_namespace=\"$namespace\",operation_type=\"unsubscribe\",servicename_3gpp=\"nudr-dr\"}[2m]))	Unsubscribe	Not Applicable
Subscribe Total UDR Request	Measures the number of subscribe requests sent by PCF to UDR.	sum (rate(ocpm_udr_tracking_request_total{kubernetes_namespace=\"$namespace\",operation_type=\"subscribe\",servicename_3gpp=\"nudr-dr\"}[2m]))	Subscribe	Not Applicable

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
Subscribe UDR Success	Measures the number of success responses received for subscribe requests sent by PCF to UDR.	sum (rate(ocpm_udr_tracking_response_total{kubernetes_namespace=\"$namespace\",operation_type=\"subscribe\",servicename_3gpp=\"nudr-dr\",response_code=\"2xx\"}[2m]))	Subscribe	2xx
Subscribe UDR Failed	Measures the number of unsuccessful responses received for subscribe requests sent by PCF to UDR.	sum (rate(ocpm_udr_tracking_response_total{kubernetes_namespace=\"$namespace\",operation_type=\"subscribe\",servicename_3gpp=\"nudr-dr\",response_code!=\"2xx\"}[2m]))	Subscribe	2xx
Query SM Data Success	Measures the number of success responses received for query requests sent by PCF to UDR.	sum (rate(ocpm_udr_tracking_response_total{kubernetes_namespace=\"$namespace\",operation_type=\"get\",servicename_3gpp=\"nudr-dr\",response_code=\"2xx\"}[2m]))	GET	2xx
Query SM Data Failed	Measures the number of unsuccessful responses received for query requests sent by PCF to UDR.	sum (rate(ocpm_udr_tracking_response_total{kubernetes_namespace=\"$namespace\",operation_type=\"get\",servicename_3gpp=\"nudr-dr\",response_code!=\"2xx\"}[2m]))	GET	2xx
Unsubscribe UDR Success	Measures the number of success responses received for unsubscribe requests sent by PCF to UDR.	sum (rate(ocpm_udr_tracking_response_total{kubernetes_namespace=\"$namespace\",operation_type=\"unsubscribe\",servicename_3gpp=\"nudr-dr\",response_code=\"2xx\"}[2m]))	Unsubscribe	2xx

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
Unsubscribe UDR Failed	Measures the number of unsuccessful responses received for unsubscribe requests sent by PCF to UDR.	sum (rate(ocpm_udr_tracking_response_total{kubernetes_namespace=\"$namespace\",operation_type=\"unsubscribe\",servicename_3gpp=\"nudr-dr\",response_code!=\"2xx\"}[2m]))	Unsubscribe	2xx
Unsubscribe UDR Timeout	Measures the number of unsubscribe requests not processed successfully at UDR due to timeout error.	sum (rate(ocpm_udr_tracking_request_timeout_total{kubernetes_namespace=\"$namespace\",operation_type=\"unsubscribe\",servicename_3gpp=\"nudr-dr\"}[2m]))	Unsubscribe	Not Applicable
Subscribe UDR Timeout	Measures the number of subscribe requests not processed successfully at UDR due to timeout error.	sum (rate(ocpm_udr_tracking_request_timeout_total{kubernetes_namespace=\"$namespace\",operation_type=\"subscribe\",servicename_3gpp=\"nudr-dr\"}[2m]))	Subscribe	Not Applicable
Query UDR Timeout	Measures the number of query requests not processed successfully at UDR due to timeout error.	sum (rate(ocpm_udr_tracking_request_timeout_total{kubernetes_namespace=\"$namespace\",operation_type=\"get\",servicename_3gpp=\"nudr-dr\"}[2m]))	GET	Not Applicable

Diameter Egress Request and Response

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
Rx RAR Request	Measures the number of requests sent to external NFs through Egress Gateway.	sum (rate(ocpm_egress_request_total{kubernetes_namespace=\"$namespace\",operation_type=\"update_notify\",servicename_3gpp=\"rx\"}[2m]))	Update Notify	Not Applicable

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
Rx RAR Response Success	Measures the number of success responses received for requests sent to external NFs through Egress Gateway.	sum (rate(ocpm_egress_response_total{kubernetes_namespace=\"$namespace\",operation_type=\"update_notify\",servicename_3gpp=\"rx\",response_code=\"2xxx\"}[2m]))	Update Notify	2xxx

User Service Inbound

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
GET Request to User Service	Measures the number of query requests received at User Service.	sum (rate(ocpm_userservice_inbound_count_total{kubernetes_namespace=\"$namespace\",operation_type=\"get\",service_resource=\"user-service\"}[2m]))	GET	Not Applicable
Delete Request to User Service	Measures the number of delete requests received at User Service.	sum (rate(ocpm_userservice_inbound_count_total{kubernetes_namespace=\"$namespace\",operation_type=\"delete\",service_resource=\"user-service\"}[2m]))	DELETE	Not Applicable
Notify Request to User Service	Measures the number of notify requests received at User Service.	sum (rate(ocpm_userservice_inbound_count_total{kubernetes_namespace=\"$namespace\",operation_type=\"notify\",service_resource=\"user-service\"}[2m]))	Notify	Not Applicable

Diameter Connector

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
Diameter Connector	Measures the total number of connections established towards Diameter Gateway.	sum (rate(occnp_diam_conn_network{kubernetes_namespace=\"$namespace\"}[2m]))	Not Applicable	Not Applicable

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
Diameter Connector	Measures the total number of connections established towards SM service application.	sum (rate(occnp_diam_conn_app_network{kubernetes_namespace=\"$namespace\"}[2m]))	Not Applicable	Not Applicable

Egress Request and Response

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
Egress Delete Request	Measures the number of delete requests sent to external NFs through Egress Gateway.	sum(rate(oc_egressgateway_http_requests_total{kubernetes_namespace=\"$namespace\",Method='DELETE'}[2m]))	DELETE	Not Applicable
Egress Delete Response	Measures the number of responses to Delete requests – sent to external NFs through Egress Gateway.	sum(rate(oc_egressgateway_http_responses_total{kubernetes_namespace=\"$namespace\",Method='DELETE'}[2m]))	DELETE	Not Applicable
Egress GET request	Measures the number of query requests sent to external NFs through Egress Gateway.	sum(rate(oc_egressgateway_http_requests_total{kubernetes_namespace=\"$namespace\",Method='GET'}[2m]))	GET	Not Applicable
Egress GET response	Measures the number of responses to query requests – sent to external NFs through Egress Gateway.	sum(rate(oc_egressgateway_http_responses_total{kubernetes_namespace=\"$namespace\",Method='GET'}[2m]))	GET	Not Applicable
Egress POST request	Measures the number of POST requests sent to external NFs through Egress Gateway.	sum(rate(oc_egressgateway_http_requests_total{kubernetes_namespace=\"$namespace\",Method='POST'}[2m]))	POST	Not Applicable
Egress POST response	Measures the number of responses to POST requests – sent to external NFs through Egress Gateway.	sum(rate(oc_egressgateway_http_responses_total{kubernetes_namespace=\"$namespace\",Method='POST'}[2m]))	POST	Not Applicable

CHF Tracking Request

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
CHF-Subscribe-request	Total number of subscribe requests (spending limit status information) from CHF server.	sum (rate(ocpm_chf_tracking_request_total{kubernetes_namespace=\"$namespace\",operation_type=\"subscribe\",servicename_3gpp=\"nchf-spendinglimitcontrol\"}[2m]))	Subscribe	Not Applicable
CHF-Unsubscribe-request	Total number of unsubscribe requests (spending limit status information) being sent to CHF server.	sum (rate(ocpm_chf_tracking_request_total{kubernetes_namespace=\"$namespace\",operation_type=\"unsubscribe\",servicename_3gpp=\"nchf-spendinglimitcontrol\"}[2m]))	Unsubscribe	Not Applicable

PolicyDS

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
Client Request Total Search	Measures the number of messages sent to LDAP gateway from PolicyDS for LDAP search requests.	sum (rate(client_request_total{kubernetes_namespace=\"$namespace\",application=\"policyds\",operation=\"SEARCH\",workflow=\"LDAP\"}[2m]))		
Client Response Total Search	Measures the number of responses sent by LDAP gateway for LDAP search requests from PolicyDS.	sum (rate(client_response_total{kubernetes_namespace=\"$namespace\",application=\"policyds\",operation=\"SEARCH\",workflow=\"LDAP\",response=\"200\"}[2m]))		

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
Server Request User Service	Measures the number of messages sent to LDAP gateway from PolicyDS for User service search requests.	sum (rate(server_request_total{kubernetes_namespace=\"$namespace\",application=\"policyds\",operation=\"SEARCH\",task=\"USER_SERVICE\"}[2m]))		
Server Request LDAP	Measures the number of messages sent to LDAP gateway from PolicyDS for LDAP retrieve requests.	sum (rate(server_request_total{kubernetes_namespace=\"$namespace\",application=\"policyds\",operation=\"GET\",task=\"LDAP\"}[2m]))		
Server Request PRE	Measures the number of messages sent to PRE from PolicyDS for inserting the user data.	sum (rate(server_request_total{kubernetes_namespace=\"$namespace\",application=\"policyds\",operation=\"INSERT\",task=\"PRE\"}[2m]))		
Server Response Success POST	Measures the number of success responses received for POST requests by PolicyDS.	sum (rate(server_response_total{kubernetes_namespace=\"$namespace\",application=\"policyds\",operation=\"POST\",response=\"200\"}[2m]))	POST	200

CHF Tracking Response

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
Subscribe Success	Measures the total number of success responses for subscribe/fetch spendingLimitStatus requests at CHF connector.	sum (rate(ocpm_chf_tracking_response_total{kubernetes_namespace=\"$namespace\",operation_type=\"subscribe\",servicename_3gpp=\"nchf-spendinglimitcontrol\",response_code=\"2xx\"}[2m]))	Subscribe	2xx

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
Subscribe Failure	Measures the total number of failed responses for subscribe/fetch spendingLimitStatus requests at CHF connector.	sum (rate(ocpm_chf_tracking_response_total{kubernetes_namespace=\"$namespace\",operation_type=\"subscribe\",servicename_3gpp=\"nchf-spendinglimitcontrol\",response_code!=\"2xx\"}[2m]))	Subscribe	2xx
Unsubscribe Success	Measures the total number of success responses for unsubscribe spendingLimitStatus requests at CHF connector.	sum (rate(ocpm_chf_tracking_response_total{kubernetes_namespace=\"$namespace\",operation_type=\"unsubscribe\",servicename_3gpp=\"nchf-spendinglimitcontrol\",response_code=\"2xx\"}[2m]))	Unsubscribe	2xx
Unsubscribe Failure	Measures the total number of failed responses for unsubscribe spendingLimitStatus requests at CHF connector.	sum (rate(ocpm_chf_tracking_response_total{kubernetes_namespace=\"$namespace\",operation_type=\"unsubscribe\",servicename_3gpp=\"nchf-spendinglimitcontrol\",response_code!=\"2xx\"}[2m]))	Unsubscribe	2xx
Unsubscribe Timeout	Measures the total number of requests that got timed out at CHF Connector when trying to unsubscribe spendingLimitStatus.	sum (rate(ocpm_chf_tracking_request_timeout_total{kubernetes_namespace=\"$namespace\",operation_type=\"unsubscribe\",servicename_3gpp=\"nchf-spendinglimitcontrol\"}[2m]))	Unsubscribe	Not Applicable

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
Subscribe Timeout	Measures the total number of requests that got timed out at CHF Connector when trying to subscribe/fetch spendingLimitStatus.	sum (rate(ocpm_chf_tracking_request_timeout_total{kubernetes_namespace=\"$namespace\",operation_type=\"subscribe\",servicename_3gpp=\"nchf-spendinglimitcontrol\"}[2m]))	Subscribe	Not Applicable

LDAP

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
LDAP Total Requests	Measures the rate of total number of requests received at LDAP.	sum (rate(ldap_request_total{kubernetes_namespace=\"$namespace\",ReqType=\"GET\"}[2m]))	All	Not Applicable
LDAP Response Success	Measures the rate of total number of successful responses sent to network NFs by LDAP.	sum (rate(ldap_response_total{kubernetes_namespace=\"$namespace\",ReqType=\"GET\",Code=\"2xx\"}[2m]))	All	2xx
LDAP Response Failure	Measures the rate of total number of requests that have been rejected by LDAP due to errors at the end of NF consumers.	sum (rate(ldap_response_total{kubernetes_namespace=\"$namespace\",ReqType=\"GET\",Code!=\"2xx\"}[2m]))	All	2xx

Policy DS

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
Client_request_total_search	Measures the sum of total number of requests policyds sends for LDAP Search.	sum (rate(client_request_total{kubernetes_namespace=\"$namespace\",application=\"policyds\",operation=\"SEARCH\",workflow=\"LDAP\"}[2m]))	SEARCH	Not Applicable

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
Client_response_total_search	Measures the sum of total number of success responses received by policyds for LDAP Search.	sum (rate(client_response_total{kubernetes_namespace=\"$namespace\",application=\"policyds\",operation=\"SEARCH\",workflow=\"LDAP\",response=\"200\"}[2m]))	SEARCH	200
Server_request_user_service	Measures the total requests received by PolicyDS from NF consumers for UserService search.	sum (rate(server_request_total{kubernetes_namespace=\"$namespace\",application=\"policyds\",operation=\"SEARCH\",task=\"USER_SERVICE\"}[2m]))	SEARCH	Not Applicable
Server_request_LDAP	Measures the total number or requests policyds received for LDAP Get request.	sum (rate(server_request_total{kubernetes_namespace=\"$namespace\",application=\"policyds\",operation=\"GET\",task=\"LDAP\"}[2m]))	GET	Not Applicable
Server_request_PRE	Measures the total number or requests policyds received for PRE Insert request.	sum (rate(server_request_total{kubernetes_namespace=\"$namespace\",application=\"policyds\",operation=\"INSERT\",task=\"PRE\"}[2m]))	INSERT	Not Applicable
Server_response_sucess_POST		sum (rate(server_response_total{kubernetes_namespace=\"$namespace\",application=\"policyds\",operation=\"POST\",response=\"200\"}[2m]))	POST	200

10.1 PCRF KPIs

This section provides information about Key Performance Indicators (KPIs) used for PCRF in CNC Policy.

> **Note**
>
> Sample PCRF dashboard for Grafana is provided in the CNC Policy Custom Templates folder as part of the documentation.

CPU, Memory, and Load Level

KPI Name	KPI Details	Metric used for KPI
Pod Health	Shows the health of pods available in the system.	appinfo_service_running{vendor=\"Oracle\", application=\"occnp\", namespace=\"$namespace\"}
POD-Count	Shows the number of PCRF pods available in the system.	count(container_memory_usage_bytes{container='pcrf-core',namespace=\"$namespace\"})
POD-Count	Shows the number of Diameter Gateway pods available in the system.	count(container_memory_usage_bytes{container='diam-gateway',namespace=\"$namespace\"})
POD-Count	Shows the number of PRE pods available in the system.	count(container_memory_usage_bytes{container='pre-service',namespace=\"$namespace\"})
POD-Count	Shows the number of Policy Datasource pods available in the system.	count(container_memory_usage_bytes{container='policyds',namespace=\"$namespace\"})
POD-Count	Shows the number of LDAP Gateway pods available in the system.	count(container_memory_usage_bytes{container='ldap-gateway',namespace=\"$namespace\"})
POD-Count	Shows the number of CM service pods available in the system.	count(container_memory_usage_bytes{container='cm-service',namespace=\"$namespace\"})
POD-Count	Shows the number of Config Server pods available in the system.	count(container_memory_usage_bytes{container='config-server',namespace=\"$namespace\"})
Memory Usage	Shows the current memory usage in bytes.	sum(container_memory_usage_bytes{image!='',namespace=\"$namespace\"})
Memory Usage	Shows the memory usage (in bytes) for the top 10 memory users by each pod.	topk(10, sum(container_memory_usage_bytes{namespace=\"$namespace\"}) by (container))
Service Load Level	Shows the load level of each service.	avg(load_level) by (service)
CPU Usage	Shows the CPU usage by each pod.	sum(rate(container_cpu_usage_seconds_total{image!='',namespace=\"$namespace\",container_name!='POD'}[2m])) by (container_name)

PCRF Success Response

KPI Name	KPI Details	Metric used for KPI	Response Code
CCA_SUCCESS_COUNT	Shows the rate of success responses received at PCRF for CCA message type.	sum(rate(occnp_diam_response_local_total{msgType=~\"CCA.*\", responseCode=~\"2.*\", namespace=\"$namespace\",container=\"pcrf-core\"}[5m]))	2xx
AAA_SUCCESS_COUNT	Shows the rate of success responses received at PCRF for AAA message type.	sum(rate(occnp_diam_response_local_total{msgType=~\"AAA.*\", responseCode=~\"2.*\", namespace=\"$namespace\",container=\"pcrf-core\"}[5m]))	2xx
ASA_SUCCESS_COUNT	Shows the rate of success responses received at PCRF for ASA message type.	sum(rate(occnp_diam_response_local_total{msgType=~\"ASA\", responseCode=~\"2.*\", namespace=\"$namespace\",container=\"pcrf-core\"}[5m]))	2xx
RAA_SUCCESS_COUNT	Shows the rate of success responses received at PCRF for RAA message type.	sum(rate(occnp_diam_response_local_total{msgType=~\"RAA\", responseCode=~\"2.*\", namespace=\"$namespace\",container=\"pcrf-core\"}[5m]))	2xx
STA_SUCCESS_COUNT	Shows the rate of success responses received at PCRF for STA message type.	sum(rate(occnp_diam_response_local_total{msgType=~\"STA\", responseCode!=\"2.*\", namespace=\"$namespace\",container=\"pcrf-core\"}[5m]))	2xx

PCRF Failure Response

KPI Name	KPI Details	Metric used for KPI	Response Code
CCA_FAIL_COUNT	Shows the rate of failure responses received at PCRF for CCA message type.	sum(rate(occnp_diam_response_local_total{msgType=~\"CCA.*\", responseCode!~\"2.*\", namespace=\"$namespace\",container=\"pcrf-core\"}[5m]))	Other than 2xx
AAA_FAIL_COUNT	Shows the rate of failure responses received at PCRF for AAA message type.	sum(rate(occnp_diam_response_local_total{msgType=~\"AAA.*\", responseCode!~\"2.*\", namespace=\"$namespace\",container=\"pcrf-core\"}[5m]))	Other than 2xx

KPI Name	KPI Details	Metric used for KPI	Response Code
ASA_FAIL_COUNT	Shows the rate of failure responses received at PCRF for ASA message type.	sum(rate(occnp_diam_response_local_total{msgType=~\"ASA\", responseCode!~\"2.*\", namespace=\"$namespace\",container=\"pcrf-core\"}[5m]))	Other than 2xx
RAA_FAIL_COUNT	Shows the rate of success responses received at PCRF for RAA message type.	sum(rate(occnp_diam_response_local_total{msgType=~\"RAA\", responseCode!~\"2.*\", namespace=\"$namespace\",container=\"pcrf-core\"}[5m]))	Other than 2xx
STA_FAIL_COUNT	Shows the rate of failure responses received at PCRF for STA message type.	sum(rate(occnp_diam_response_local_total{msgType=~\"STA\", responseCode!~\"2.*\", namespace=\"$namespace\",container=\"pcrf-core\"}[5m]))	Other than 2xx

PCRF Timeout Failure

KPI Name	KPI Details	Metric used for KPI
ASA_TIMEOUT_COUNT	Shows the rate of failure responses due to timeout for ASA message type.	sum(rate(occnp_diam_response_local_total{msgType=~\"ASA\", responseCode=\"timeout\", namespace=\"$namespace\",container=\"pcrf-core\"}[5m]))
RAA_TIMEOUT_COUNT	Shows the rate of failure responses due to timeout for RAA message type.	sum(rate(occnp_diam_response_local_total{msgType=~\"RAA\", responseCode=\"timeout\", namespace=\"$namespace\",container=\"pcrf-core\"}[5m]))

Diameter Gateway Timeout Failure

KPI Name	KPI Details	Metric used for KPI
CCA_TIMEOUT_COUNT	Shows the rate of failure responses due to timeout received at Diameter Gateway for CCA message type.	sum(rate(occnp_diam_response_local_total{msgType=~\"CCA.*\", responseCode=\"timeout\", namespace=\"$namespace\",container=\"diam-gateway\"}[5m]))
AAA_TIMEOUT_COUNT	Shows the rate of failure responses due to timeout received at Diameter Gateway for AAA message type.	sum(rate(occnp_diam_response_local_total{msgType=~\"AAA.*\", responseCode=\"timeout\", namespace=\"$namespace\",container=\"diam-gateway\"}[5m]))
STA_TIMEOUT_COUNT	Shows the rate of failure responses due to timeout received at Diameter Gateway for STA message type.	sum(rate(occnp_diam_response_local_total{msgType=~\"STA\", responseCode=\"timeout\", namespace=\"$namespace\",container=\"diam-gateway\"}[5m]))

Diameter Gateway Response

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response Code
FAILURE_RATE	Shows the percentage of total failure responses of all incoming messages (for Diameter Gateway and PCRF) per unit time.	sum(rate(occnp_diam_response_local_total{namespace=\"$namespace\",responseCode!~\"2.*\"}[5m])) / sum(rate(occnp_diam_response_local_total{namespace=\"$namespace\"}[5m]))*100	Not Applicable	Other than 2xx
SUCCESS_RATE	Shows the percentage of total success responses of all incoming messages (for Diameter Gateway and PCRF) per unit time.	sum(rate(occnp_diam_response_local_total{namespace=\"$namespace\",responseCode=~\"2.*\"}[5m])) / sum(rate(occnp_diam_response_local_total{namespace=\"$namespace\"}[5m]))*100	Not Applicable	2xx

A

Ingress Gateway Metrics

Ingress Gateway Metrics

Ingress Metrics Common Tags

Tags	Description	Possible Values
Method	Http method.	• GET • PUT • POST • DELETE • PATCH
NFType	Name of the NF Type.	For Eg: Path is /n**xxx-yyy**/v**z**/..... Where **XXX(Upper Case)** is NFType UNKNOWN if unable to extract NFType from the path
NFServiceType	Name of the Service with in the NF.	For Eg: Path is /n**xxx-yyy**/v**z**/..... Where nxxx-yyy is NFServiceType UNKNOWN if unable to extract NFServiceType from the path
Host	Port of ingress gateway (Ip or fqdn).	NA
HttpVersion	Http protocol version.	• HTTP/1.1 • HTTP/2.0
Scheme	Http protocol scheme.	HTTP, HTTPS, UNKNOWN
ClientCertIdentity	Cerificate Identity of the client.	SAN=127.0.0.1,localhost CN=localhost, N/A if data is not available
Route_Path	Path predicate/Header predicate that matched the current request.	NA
InstanceIdentifier	Prefix of the pod configured in helm when there are multiple instances in same deployment.	Prefix configured in helm otherwise UNKNOWN
ErrorOriginator	This tag captures the ErrorOriginator.	ServiceProducer, Nrf, IngresGW, None
oc_ingressgateway_route_ratelimit_**Status** oc_ingressgateway_global_ratelimit_**Status**	Request accepted or dropped.	accepted, dropped
oc_ingressgateway_connection_failure_**Host**	Destination ip/fqdn.	NA
oc_ingressgateway_connection_failure_**Port**	Destination port.	NA
oc_ingressgateway_xfcc_header_validate_**Status**	Https Status value after performing xfccHeaderValidation at Ingress Gateway.	200 (OK), 400 (BAD_REQUEST)

Tags	Description	Possible Values
oc_ingressgateway_xfcc_header_validate_**Cause**	This tag determines the validation cause for the xfcc header validation metric being pegged.	VALIDATION_FAILURE, VALIDATION_SUCCESS, HEADER_NOT_FOUND
oc_ingressgateway_xfcc_header_validate_**CertsCompared**	This tag captures the total number of certificates compared in XFCC header at ingress gateway during the header validation.	Count of the certificates compared (0,1,2..)
oc_configclient_request_total_**releaseVersion**	This tag indicates the current release version of ingress gateway.	Picked from helm chart{{ .Chart.Version }}
oc_configclient_request_total_**configVersion**	This tag indicates the configuration version that ingress gateway is currently maintaining.	Initial value is 0. Incremental value received from config server whenever there is an update from config server (0, 1, 2...)
oc_configclient_response_total_**releaseVersion**	This tag indicates the current release version of ingress gateway.	Picked from helm chart {{ .Chart.Version }}
oc_configclient_response_total_**configVersion**	This tag indicates the configuration version that ingress gateway is currently maintaining.	Value received from config server (1, 2...)
oc_configclient_response_total_**updated**	This tag indicates whether the configuration was updated or not.	true/false

Ingress Gateway Metrics

Table A-1 oc_ingressgateway_http_requests_total

Field	Details
Description	This metric is pegged as soon as the request reaches the Ingress gateway in the first custom filter of the application.
Type	Counter
Dimension	• NFType • NFServiceType • Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • ClientCertIdentity

Table A-2 oc_ingressgateway_http_responses_total

Field	Details
Description	This metric is pegged in the last custom filter of the Ingress gateway while the response is being sent back to the consumer NF.
Type	Counter

Table A-2 (Cont.) oc_ingressgateway_http_responses_total

Field	Details
Dimension	• Status • Method • Route_path • NFType • NFServiceType • Host • HttpVersion • Scheme • Identifier • ClientCertIdentity

Table A-3 oc_ingressgateway_request_latency_seconds

Field	Details
Description	This metric is pegged in the last custom filter of the Ingress gateway while the response is being sent back to the consumer NF. This metric tracks the amount of time taken for processing the request. It starts as soon the request reaches the first custom filter of the application and lasts till the response is sent back to the consumer NF from the last custom filter of the application.
Type	Timer
Dimension	• quantile • InstanceIdentifier

Table A-4 oc_ingressgateway_connection_failure_total

Field	Details
Description	This metric is pegged in the customized Jetty Client as soon as it fails to connect to the destination service with direction as ingressOut. Here in case of Ingress gateway, the destination service is a backend microservice of the NF. And TLS connection failure metrics when connecting to ingress with direction as ingress.
Type	Counter
Dimension	• Host • Port • Direction • InstanceIdentifier • error_reason

Table A-5 oc_ingressgateway_global_ratelimit_total

Field	Details
Description	This metric is pegged in the custom filter implemented to check the global rate limit conditions.
Type	Counter

Table A-5 (Cont.) oc_ingressgateway_global_ratelimit_total

Field	Details
Dimension	• Method • Route_path • Scheme • InstanceIdentifier • Status (Rate limit Status field is different here)

Table A-6 oc_ingressgateway_route_ratelimit_total

Field	Details
Description	This metric is pegged in the custom filter implemented to check the route level rate limit conditions.
Type	Counter
Dimension	• Method • Route_path • Scheme • InstanceIdentifier • Status (Rate limit Status field is different here)

Table A-7 oc_ingressgateway_request_processing_latency_seconds

Field	Details
Description	This metric is pegged in the last custom filter of the Ingress gateway while the response is being sent back to the consumer NF. This metric captures the amount of time taken for processing of the request only within Ingress gateway. It starts as soon the request reaches the first custom filter of the application and lasts till the request is forwarded to the destination.
Type	Timer
Dimension	• quantile • InstanceIdentifier

Table A-8 oc_ingressgateway_jetty_request_stat_metrics_total

Field	Details
Description	This metric is pegged for every event occurred when a request is sent to IGW.
Type	Counter
Dimension	• event • client_type • InstanceIdentifier

Table A-9 oc_ingressgateway_jetty_response_stat_metrics_total

Field	Details
Description	This metric is pegged for every event occurred when a response is received by IGW
Type	Counter

Table A-9 (Cont.) oc_ingressgateway_jetty_response_stat_metrics_total

Field	Details
Dimension	• event • client_type • InstanceIdentifier

Table A-10 oc_ingressgateway_jetty_latency_seconds

Field	Details
Description	This metric is pegged in Jetty response listener that captures the amount of time taken for processing of the request by jetty client.
Type	Timer
Dimension	• quantile • InstanceIdentifier

Table A-11 oc_ingressgateway_netty_latency_seconds

Field	Details
Description	This metric is pegged in Netty outbound handler that captures the amount of time taken for processing of the request by netty server.
Type	Timer
Dimension	• quantile • InstanceIdentifier

Table A-12 oc_ingressgateway_request_content_metrics_total

Field	Details
Description	This metric is pegged by default filter RequestContentMetrics. It pegs whether request has request body or not.
Type	Counter
Dimension	• method • content_available • InstanceIdentifier

Table A-13 oc_ingressgateway_xfcc_header_validate_total

Field	Details
Description	This metric is pegged when xfccHeaderValidation is enabled in XfccHeaderValidationFilter. This metric along with the specified dimension captures the successful/ un-successful validation of XFCC header in the incoming request.
Type	Counter

Table A-13 (Cont.) oc_ingressgateway_xfcc_header_validate_total

Field	Details
Dimension	• Route_path • Status • Cause • CertsCompared • InstanceIdentifier • ErrorOriginator

Table A-14 oc_configclient_request_total

Field	Details
Description	This metric is pegged whenever config client is polling for configuration update from common configuration server.
Type	Counter
Dimension	• Release version • Config version

Table A-15 oc_configclient_response_total

Field	Details
Description	This metrics is pegged whenever config client receives response from common configuration server.
Type	Counter
Dimension	• Release version • Config version • Updated

Oauth Metrics

OAuth Metrics Common Tags

Tags	Description	Possible Values
scope	NF service name(s) of the NF service producer(s), separated by white spaces.	NA
issuer	NF instance id of NRF.	NA
subject	NF instance id of service consumer.	NA
reason	reason contains the human readable message for oauth validation failure.	NA

Below are the metrics and their respective tags that are available in Oauth:

Table A-16 oc_oauth_validation_successful _total

Field	Details
Description	This metric is pegged in the OAuth validator implementation if the received OAuth token is validated successfully. The implementation of OAuth validator is used in Ingress Gateway.
Type	Counter
Dimension	• issuer • subject • scope

Table A-17 oc_oauth_validation_failure _total

Field	Details
Description	This metric is pegged in the implementation of OAuth validator if the validation of the the received OAuth token fails. The implementation of OAuth validator is used in Ingress G;ateway.
Type	Counter
Dimension	• issuer • subject • scope • reason

Table A-18 oc_ingressgateway_msgcopy_requests_total

Field	Details
Description	This is incremented whenever ingress request message is sent or acknowledged from Kafka.
Type	Counter
Dimension	
Example	

Table A-19 oc_ingressgateway_msgcopy_responses_total

Field	Details
Description	This is incremented whenever ingress response message is sent or acknowledged from Kafka.
Type	Counter
Dimension	
Example	

B
Egress Gateway Metrics

Egress Gateway Metrics

The following table describes the Egress Gateway Metrics.

Table B-1 oc_egressgateway_http_requests_total

Field	Details
Available Tags	• Method • NFType • NFServiceType • Host • HttpVersion • Scheme • Proxy • InstanceIdentifier
Pegging Instance	This metric is pegged as soon as the request reaches the Egress Gateway in the first custom filter of the application.

Table B-2 oc_egressgateway_http_responses_total

Field	Details
Available Tags	• Status • Method • NFType • NFServiceType • Host • HttpVersion • Scheme • InstanceIdentifier • Direction • BlacklistedFqdn
Pegging Instance	This metric will be pegged in the last custom filter of the Egress gateway while the response is being sent back to backend NF microservice with direction as egress. This will also be pegged when the response is fetched in Jetty responseListener with direction as egressOut. BlacklistedFqdn tag will be filled with BlacklistedFqdn when request is sent with blacklisted producer.

Table B-3 oc_egressgateway_request_latency_seconds

Field	Details
Available Tags	• quantile • InstanceIdentifier
Pegging Instance	This metric is pegged in the last custom filter of the Ingress Gateway while the response is being sent back to the consumer NF. This metric tracks the amount of time taken for processing the request. It starts as soon as the request reaches the first custom filter of the application and lasts till, the response is sent back to the the consumer NF from the last custom filter of the application.

Table B-4 oc_egressgateway_connection_failure_total

Field	Details
Available Tags	• Host • Port • InstanceIdentifier • Direction • error_reason
Pegging Instance	This metric will be pegged in the customized Jetty Client as soon as it fails to connect to the destination service. Here in case of Egress gateway, the destination service will be Producer NF. This will also be pegged when the request to Producer NF fails in Jetty request Listener with direction as egressOut

Table B-5 oc_egressgateway_notification_ratelimit_total

Field	Details
Available Tags	• Method • Scheme • InstanceIdentifier
Pegging Instance	This metric is pegged in the custom filter implemented to check the notification rate limit conditions.

Table B-6 oc_egressgateway_request_processing_latency_seconds

Field	Details
Available Tags	• quantile • InstanceIdentifier

Table B-6 (Cont.) oc_egressgateway_request_processing_latency_seconds

Field	Details
Pegging Instance	This metric is pegged in the last custom filter of the Egress Gateway while the response is sent back to the consumer NF. This metric tracks the amount of time taken for processing the request only within Egress Gateway. It starts as soon as the request reaches the first custom filter of the application and lasts till the request is forwarded to the destination.

Table B-7 oc_egressgateway_jetty_request_stat_metrics_total

Field	Details
Available Tags	• event • client_type • InstanceIdentifier
Pegging Instance	This metric is pegged for every event occurred when a request is sent to EGW

Table B-8 oc_egressgateway_jetty_response_stat_metrics_total

Field	Details
Available Tags	• event • client_type • InstanceIdentifier
Pegging Instance	This metric is pegged for every event occurred when a response is received by EGW

Table B-9 oc_egressgateway_jetty_response_stat_metrics_total

Field	Details
Available Tags	• event • client_type • InstanceIdentifier
Pegging Instance	This metric is pegged for every event occurred when a response is received by EGW

Table B-10 oc_egressgateway_jetty_latency_seconds

Field	Details
Available Tags	• quantile • InstanceIdentifier
Pegging Instance	This metric is pegged in Jetty response listener that captures the amount of time taken for processing of the request by jetty client

Table B-11 oc_egressgateway_jetty_latency_seconds

Field	Details
Available Tags	• quantile • InstanceIdentifier
Pegging Instance	This metric is pegged in Jetty response listener that captures the amount of time taken for processing of the request by jetty client

Table B-12 oc_egressgateway_netty_latency_seconds

Field	Details
Available Tags	• quantile • InstanceIdentifier
Pegging Instance	This metric is pegged in Netty outbound handler that captures the amount of time taken for processing of the request by netty server

Table B-13 oc_egressgateway_request_content_metrics_total

Field	Details
Available Tags	• method • content_available • InstanceIdentifier
Pegging Instance	This metric is pegged by default filter RequestContentMetrics. It pegs whether request has request body or not and the method.

Table B-14 oc_egressgateway_blacklisted_producer_total

Field	Details
Available Tags	• NFType • NFServiceType • InstanceIdentifier • Host • Route_path
Pegging Instance	This metric is a counter. Track number of times producer is blacklisted.

Table B-15 oc_configclient_request_total

Field	Details
Available Tags	• Release version • Config version
Pegging Instance	This metric will be pegged whenever config client is polling for configuration update from common configuration server

Table B-16 oc_configclient_response_total

Field	Details
Available Tags	• Release version • Config version • Updated
Pegging Instance	This metrics will be pegged whenever config client receives response from common configuration server

Egress Gateway Metrics Common Tags

The following table describes the common tags used in Egress Gateway Metrics.

Table B-17 Method

Field	Details
Available Tags	Http method
Pegging Instance	GET, PUT, POST, DELETE, PATCH

Table B-18 NFType

Field	Details
Available Tags	Name of the NF Type
Pegging Instance	"UNKNOWN" (Updates are available when Ingress is 5G aware)

Table B-19 NFServiceType

Field	Details
Available Tags	Name of the Service within the NF
Pegging Instance	"UNKNOWN" (Updates are available when Ingress is 5G aware)

Table B-20 Host

Field	Details
Available Tags	(IP or fqdn): port of ingress gateway
Pegging Instance	Not Applicable

Table B-21 HttpVersion

Field	Details
Available Tags	Http protocol version (http1.1/ http2)
Pegging Instance	HTTP1.1, HTTP2.0

Table B-22 Scheme

Field	Details
Available Tags	Http protocol scheme (http/https)
Pegging Instance	HTTP, HTTPS, UNKNOWN

Table B-23 Proxy

Field	Details
Available Tags	Value received for "x-custom-egress-proxy-header".
Pegging Instance	Unknown or value of "x-custom-egress-proxy-header".

Table B-24 oc_egressgateway_connection_failure_Host

Field	Details
Available Tags	destination ip/fqdn
Pegging Instance	Not Applicable

Table B-25 oc_egressgateway_connection_failure_Port

Field	Details
Available Tags	destination port
Pegging Instance	Not Applicable

Table B-26 BlacklistedFqdn

Field	Details
Available Tags	Blacklisted Producer Fqdn
Pegging Instance	Unknown or Blacklisted Producer Fqdn

Table B-27 oc_configclient_request_total_releaseVersion

Field	Details
Available Tags	This tag indicates the current release version of egress gateway
Pegging Instance	Picked from helm chart{{ .Chart.Version }}

Table B-28 oc_configclient_request_total_configVersion

Field	Details
Available Tags	This tag indicates the configuration version that egress gateway is currently maintaining

Table B-28 (Cont.) oc_configclient_request_total_configVersion

Field	Details
Pegging Instance	Initial value is 0. Incremental value received from config server whenever there is an update from config server (0, 1, 2...)

Table B-29 oc_configclient_response_total_releaseVersion

Field	Details
Available Tags	This tag indicates the current release version of egress gateway
Pegging Instance	Picked from helm chart{{ .Chart.Version }}

Table B-30 oc_configclient_response_total_configVersion

Field	Details
Available Tags	This tag indicates the configuration version that egress gateway is currently maintaining
Pegging Instance	Value received from config server (1, 2...)

Table B-31 oc_configclient_response_total_updated

Field	Details
Available Tags	This tag indicates whether the configuration was updated or not
Pegging Instance	true/false

SCP Metrics

The following table describes the different metrics and their respective tags that are available in the SCP Module:

Table B-32 oc_egressgateway_scp_http_requests_total

Field	Details
Available Tags	• Scp_Fqdn • Reroute_Path • Response_Code (This would be populated as blank for requests) • Attempt • HttpVersion • Scheme • InstanceIdentifier
Pegging Instance	This metric is pegged in the ScpFilter only when SCP Integration is enabled.

Table B-33 oc_egressgateway_scp_http_responses_total

Field	Details
Available Tags	• Scp_Fqdn • Reroute_Path • Response_Code • Attempt • HttpVersion • Scheme • InstanceIdentifier
Pegging Instance	This metric is pegged in the ScpFilter only when Scp Integration is enabled. It is also being pegged in the Scp Retry Filter when Scp re-route feature is enabled.

SCP Metrics common tags

The following table describes common tags used in SCP Metrics.

Table B-34 Scp_Fqdn

Field	Details
Description	SCP Fqdn
Possible Values	Not Applicable

Table B-35 Reroute_Path

Field	Details
Description	Path that matched the request to over corresponding route Example: `/nef/**`
Possible Values	Not Applicable

Table B-36 Response_Code

Field	Details
Description	It is populated as blank for request metrics. During failure scenario's, it is populated with "ERROR" for response metrics. Example: ERROR, OK
Possible Values	ERROR, NOT ACCEPTABLE, OK

Table B-37 Attempt

Field	Details
Description	Attempt number for scp re-route. Example: 1, 2 etc.
Possible Values	Not Applicable

Table B-38 HttpVersion

Field	Details
Description	Http protocol version (http1.1/ http2)
Possible Values	HTTP/1.1, HTTP/2.0

Table B-39 Scheme

Field	Details
Description	Http protocol scheme (http/https)
Possible Values	HTTP, HTTPS, UNKNOWN

Table B-40 InstanceIdentifier

Field	Details
Description	Prefix of the pod configured in helm when there are multiple instances in same deployment
Possible Values	Prefix configured in helm otherwise UNKNOWN

Oauth Metrics

The following table includes metrics and their respective tags that are available in the Oauth Module:

Table B-41 oc_oauth_nrf_request_total

Field	Details
Available Tags	• ConsumerNFInstanceId • ConsumerNFType • TargetNFType • TargetNFInstanceId • scope • NrfFqdn
Pegging Instance	This metric is pegged in the OAuth client implementation if the request is sent to NRF for requesting the OAuth token. OAuth client implementation will be used in Egress gateway.

Table B-42 oc_oauth_nrf_response_success_total

Field	Details
Available Tags	• ConsumerNFInstanceId • ConsumerNFType • TargetNFType • TargetNFInstanceId • scope • StatusCode • NrfFqdn

Table B-42 (Cont.) oc_oauth_nrf_response_success_total

Field	Details
Pegging Instance	This metric is pegged in the OAuth client implementation if an OAuth token is successfully received from the NRF. OAuth client implementation is used in the Egress Gateway.

Table B-43 oc_oauth_nrf_response_failure_total

Field	Details
Available Tags	• ConsumerNFInstanceId • ConsumerNFType • TargetNFType • TargetNFInstanceId • scope • StatusCode • ErrorOriginator • NrfFqdn
Pegging Instance	This metric is pegged in the OAuthClientFilter in Egress Gateway whenever GetAccessTokenFailedException is caught.

Table B-44 oc_oauth_request_failed_internal_total

Field	Details
Available Tags	• ConsumerNFInstanceId • ConsumerNFType • TargetNFType • TargetNFInstanceId • scope • StatusCode • ErrorOriginator • NrfFqdn
Pegging Instance	This metric is pegged in the OAuthClientFilter in Egress Gateway whenever InternalServerErrorException is caught.

Table B-45 oc_oauth_token_cache_total

Field	Details
Available Tags	• ConsumerNFInstanceId • ConsumerNFType • TargetNFType • TargetNFInstanceId • scope
Pegging Instance	This metric is pegged in the OAuth Client Implementation if the OAuth token is found in the cache.

Table B-46 oc_oauth_request_invalid_total

Field	Details
Available Tags	• ConsumerNFInstanceId • ConsumerNFType • TargetNFType • TargetNFInstanceId • scope • StatusCode • ErrorOriginator
Pegging Instance	This metric is pegged in the OAuthClientFilter in Egress Gateway whenever a BadAccessTokenRequestException/ JsonProcessingException is caught.

Table B-47 oc_egressgateway_ouath_access_token_request_header_missing

Field	Details
Available Tags	NA
Pegging Instance	This metric is pegged in the OAuthClientFilter in Egress Gateway whenever oc-access-token-request-info header is missing in the request.

Table B-48 oc_oauth_cert_expiryStatus

Field	Details
Available Tags	• id • certificateName • secretName
Pegging Instance	This Gauge metric is used to peg expiry date of the certificate. This metric is further used for raising alarms if certificate expires within 30 days or 7 days.

Table B-49 oc_oauth_cert_loadStatus

Field	Details
Available Tags	• id • certificateName • secretName
Pegging Instance	This gauge metric is used to peg expiry date of the certificate. This metric is further used for raising alarms if certificate expires within 30 days or 7 days.

Table B-50 oc_oauth_request_failed_cert_expiry

Field	Details
Available Tags	• target nf type • target nf instance id • consumer nf instance id • nrf instance id • service name of nf producer service • key id
Pegging Instance	This counter metric is used to keep track of number of requests with keyId in token that failed due to certificate expiry. It is pegged whenever oAuth Validator module throws oauth custom exception due to certificate expiry for an incoming request.

Table B-51 oc_oauth_keyid_count

Field	Details
Available Tags	• target nf type • target nf instance id • consumer nf instance id • nrf instance id • service name of nf producer service • key id
Pegging Instance	This counter metric used to keep track of number of requests received with keyId in token. It is pegged whenever a request with an access token containing keyid in header comes to oAuth Validator.

Table B-52 oc_oauth_nrf_token_retrieval_failure_total

Field	Details
Available Tags	• ConsumerNFInstanceId • ConsumerNFType • TargetNFType • TargetNFInstanceId • scope • StatusCode • ErrorOriginator • ErrorDetail • NrfFqdn
Pegging Instance	This metric is pegged to track requests discarded due to oAuth token retrieval failure from NRF.

OAuth Metrics (NRF-Client Mgmt Service Call-Flow)

Table B-53 oc_oauth_nrf_client_subscription_request_total

Field	Details
Available Tags	• NrfClientUrl • EgwNotificationUrl
When it is pegged	This metric will be pegged in the OAuth client implementation module when a subscription request is sent from EGW to NRF-Client Mgmt Svc with request-URL (NrfClientUrl) and request body containing notification-URL of EGW (EgwNotificationUrl) for Oauth Client notification requests generated from NRF-Client Mgmt Svc to EGW. OAuth client implementation will be used in Egress gateway.

Table B-54 oc_oauth_nrf_client_notification_request_total

Field	Details
Available Tags	None
When it is pegged	This metric will be pegged in the OAuth client implementation module when a notification request is sent from NRF-Client Mgmt Svc to EGW.OAuth client implementation will be used in Egress gateway.

Table B-55 oc_oauth_nrf_client_subscription_response_total

Field	Details
Available Tags	• NrfClientUrl • EgwNotificationUrl • StatusCode
When it is pegged	This metric will be pegged in the OAuth client implementation module when a subscription response is sent from NRF-Client Mgmt Svc having URL (NrfClientUrl) to EGW having URL (EgwNotificationUrl). StatusCode tag will capture the corresponding response status obtained from NRF-Client. OAuth client implementation will be used in Egress gateway.

Table B-56 oc_oauth_nrf_client_notification_response_total

Field	Details
Available Tags	• StatusCode

Table B-56 (Cont.) oc_oauth_nrf_client_notification_response_total

Field	Details
When it is pegged	This metric will be pegged in the OAuth client implementation modulewhen a notification response is sent from EGW to NRF-Client Mgmt Svc.StatusCode tag will capture the corresponding response status sent fromEGW. OAuth client implementation will be used in Egress gateway.

Table B-57 oc_oauth_nrf_client_active_nrf_instances

Field	Details
Available Tags	• NrfFqdn
When it is pegged	This is a GAUGE metric which keeps track of healthy NRF Fqdns recievedfrom NRF-Client Mgmt Svc as part of subscription response/ notificationrequest to EGW.

OAuth Metrics common tags

The following table describes common tags used in the OAuth Module.

Table B-58 ConsumerNFInstanceId

Field	Details
Description	NF instance id of the NF service consumer
Possible Values	Not Applicable

Table B-59 ConsumerNFType

Field	Details
Description	The NF type of the NF service consumer
Possible Values	NRF, UDM, AMF, SMF, AUSF, NEF, PCF, SMSF, NSSF, UDR, LMF, GMLC,5G_EIR, SEPP, UPF, N3IWF, AF, UDSF, BSF, CHF, NWDAF

Table B-60 TargetNFType

Field	Details
Description	The NF type of the NF service producer
Possible Values	NRF, UDM, AMF, SMF, AUSF, NEF, PCF, SMSF, NSSF, UDR, LMF, GMLC,5G_EIR, SEPP, UPF, N3IWF, AF, UDSF, BSF, CHF, NWDAF

Table B-61 TargetNFInstanceId

Field	Details
Description	NF instance id of the NF service producer

Table B-61 (Cont.) TargetNFInstanceId

Field	Details
Possible Values	Not Applicable

Table B-62 scope

Field	Details
Description	NF service name(s) of the NF service producer(s), separated by whitespaces
Possible Values	Not Applicable

Table B-63 StatusCode

Field	Details
Description	Status code of NRF access token request
Possible Values	Bad Request, Internal Server Error etc. (HttpStatus.*)

Table B-64 ErrorOriginator

Field	Details
Description	from where error is originated (nrf or egress)
Possible Values	Nrf, EgressGW

Table B-65 issuer

Field	Details
Description	NF instance id of NRF
Possible Values	Not Applicable

Table B-66 subject

Field	Details
Description	NF instance id of service consumer
Possible Values	Not Applicable

Table B-67 reason

Field	Details
Description	reason contains the human readable msg for oauth validation failure
Possible Values	Not Applicable

Table B-68 NrfFqdn

Field	Details
Description	NrfFqdn tag determines the corresponding fqdn of NRF where the request has been forwarded to.
Possible Values	Nrf-Fqdn (dynamic value based on Fqdn), NA

Table B-69 NrfClientUrl

Field	Details
Description	This tag determines the url of NRF-Client Mgmt Svc where subscription requests are sent from OAuth Client module in EGW.
Possible Values	URL of NRF-Client Mgmt Svc (Dynamic value)

Table B-70 EgwNotificationUrl

Field	Details
Description	This tag determines the notification URL mapped in OAuth Client module of EGW where NRF-Client Mgmt Svc will send notifications requests.
Possible Values	Notification URL (Dynamic value)

Table B-71 ConfigurationType

Field	Details
Description	This tag determines the type of configuration in place for OAuth Client in Egress Gateway. If nrfClientQueryEnabled Helm parameter in oauthClient Helm configurations at Egress Gateway is false then the ConfigurationType is STATIC, else DYNAMIC.
Possible Values	STATIC, DYNAMIC

Table B-72 oc_egressgateway_msgcopy_requests_total

Field	Details
Description	This is incremented whenever egress request message is sent or acknowledged from Kafka.
Type	Counter
Dimension	
Example	

Table B-73 oc_egressgateway_msgcopy_responses_total

Field	Details
Description	This is incremented whenever egress response message is sent or acknowledged from Kafka.

Table B-73 (Cont.) oc_egressgateway_msgcopy_responses_total

Field	Details
Type	Counter
Dimension	
Example	

C
NRF Client Metrics

NRF Client Management Service Metrics

Table C-1

S. No	Metric Name	Metric filter	Dimensions	Notes
1	NRF Instance Status	oc_nrfclient_nrf_operative_status	NrfUri - URI of the NRF Instance	If the metric has value 0 - NRF is unavailable/ unhealthy 1 - NRF is available/healthy
2	NRF Instance Status Count	oc_nrfclient_nrf_status_total	NrfUri - URI of the NRF Instance HealthStatus FailureReason - Reason for the status	apiRoot shall be specified in the following format: *'scheme'://'fqdn':'port'* If health check procedure is disabled, all NRF instances are marked as HEALTHY after successful NfRegistration.
3	NRF Instance Consecutive Healthy Count	oc_nrfclient_nrf_successive_healthy_count	NrfUri - URI of the NRF Instance	The metric shall have a minimum value of 0 and maximum value of healthCheckCount.
4	NRF Instance Consecutive Unhealthy Count	oc_nrfclient_nrf_successive_unhealthy_count	NrfUri - URI of the NRF Instance	The metric shall have a minimum value of 0 and maximum value of healthCheckCount.
5	DNS lookup requests	oc_nrfclient_dns_lookup_request_total	Scheme - http or https VirtualFqdn - Fqdn that shall be used by the alternate service for the DNS lookup.	The metric shall be pegged only if enableVirtualNrfResolution is set to true.

Table C-1 (Cont.)

S. No	Metric Name	Metric filter	Dimensions	Notes
6	DNS lookup responses	oc_nrfclient_dns_lookup_response_total	Scheme - http or https VirtualFqdn - Fqdn that shall be used by the alternate service for the DNS lookup. HttpStatusCode - The status code as received in the response.	The metric shall be pegged only if enableVirtualNrfResolution is set to true.
7	DNS setup requests	oc_nrfclient_dns_register_request_total	Scheme - http or https VirtualFqdn - Fqdn that shall be used by the alternate service for the DNS lookup.	The metric shall be pegged only if enableVirtualNrfResolution is set to true.

D

HTTP Error Codes Supported by Policy

HTTP Error Codes supported by Policy for AM, SM, and UE Interfaces

Table D-1 HTTP Error codes and cause supported by Policy for AM, SM, and UE Interfaces for Notification

Error Code	Message	Cause Code	Description	Action
400	Bad requests	INVALID_MSG_FORMAT	Tthe HTTP request has an invalid format.	• POST retry with POST request • If retry is exhausted, then terminate the transaction.
		MANDATORY_IE_INCORRECT	A mandatory IE for an HTTP method received a semantically incorrect request.	
		MANDATORY_IE_MISSING	A mandatory IE for an HTTP method is not included in the request.	
		UNSPECIFIED_MSG_FAILURE	The request is rejected due to unspecified client error.	
		ERROR_REQUEST_PARAMETERS	The HTTP request is rejected because the set of information needed by the PCF for UE Policy selection is incomplete for the decision to be made.	
401	Unauthorized	With header "WWW-Authenticate"	There will be a new function to enable retry for this error code.	• POST retry with POST request • If retry is exhausted, terminate the transaction
403	Forbidden	MODIFICATION_NOT_ALLOWED	The request is rejected because modification is not allowed.	• POST retry with POST request • If retry is exhausted, terminate the transaction
404	Not Found	SUBSCRIPTION_NOT_FOUND	The request for modification or deletion of subscription is rejected because the subscription is not found in the NF.	• POST retry with POST request • If retry is exhausted, delete the session
		RESOURCE_URI_STRUCTURE_NOT_FOUND	The request is rejected because a fixed part after the first variable part of an "apiSpecificResourceUriPart" is not found in the NF.	• POST retry with POST request • If retry is exhausted, delete the session

Table D-1 (Cont.) HTTP Error codes and cause supported by Policy for AM, SM, and UE Interfaces for Notification

Error Code	Message	Cause Code	Description	Action
405	Method Not Allowed	NA	If the NF supports the HTTP method for several resources in the API, but not for the target resource of a given HTTP request	• POST retry with POST request • If retry is exhausted, terminate the transaction
406	Not Acceptable	NA	The server cannot generate a response in a format specified as acceptable by the client in the "Accept" headers.	
408	Request Timeout	NA	The server did not receive a complete request from the client within the expected timeframe.	
409	Conflict	NA	The server did not receive a complete request from the client within the expected timeframe.	
410	Gone	NA	The server did not receive a complete request from the client within the expected timeframe.	
411	Length Required	INCORRECT_LENGTH	The request is rejected due to incorrect valueof a Content-length header field.	
412	Precondition Failed	NA	One or more preconditions specified in the request headers were not met.	
413	Payload Too Large	NA	The server refuses to process the request because thepayload size exceeds its limitations.	
414	URI Too Long	NA	The server cannot process the request because the provided URI exceeds its maximum length limit.	
415	Unsupported Media Type	NA	The server cannot process the request due to an unsupported media type in the request payload.	

Table D-1 (Cont.) HTTP Error codes and cause supported by Policy for AM, SM, and UE Interfaces for Notification

Error Code	Message	Cause Code	Description	Action
429	Too Many Requests	NF_CONGESTION_RISK	The request is rejected due to excessive traffic which, if continued over time, may lead to (or may increase) an overload situation.	
500		INSUFFICIENT_RESOURCES	The request is rejected due to insufficient resources.	
		UNSPECIFIED_NF_FAILURE	The request is rejected due to unspecified reason at the NF.	
		SYSTEM_FAILURE	The request is rejected due to generic error condition in the NF.	
		NF_FAILOVER	The request is rejected due to generic error condition in the NF.	
503		NF_CONGESTION	The NF experiences congestion and performs overload control, which does not allow the request to be processed.	

Policy as a producer of Error to AM Interface

Table D-2 Policy as a producer of Error to AM Interface

Error Code	Message	Cause Code	Scenario
400	Bad request	MANDATORY_IE_INCORRECT	When the SUPI was incorrectly written into the request body and sent to Ingress.
		MANDATORY_IE_MISSING	When a request is sent without a mandatory IE (SUPI) in the request body.
		UNSPECIFIED_MSG_FAILURE	When there is malformed JSON object.
		USER_UNKNOWN	A correct request is sent with a SUPI that is not registered in the UDR. The validate-user in the CNC Console of the UE Policy is activated.
		ERROR_REQUEST_PARAMETERS	When there is malformed JSON object.
		PENDING_TRANSACTION	In case of Pending transaction.
401	Unauthorized	With header "WWW-Authenticate"	Supported by Gateway service.

Table D-2 (Cont.) Policy as a producer of Error to AM Interface

403	Forbidden	MODIFICATION_NOT_ALLOWED	Supported by Gateway service.
404	Not Found	SUBSCRIPTION_NOT_FOUND	When UE-Delete is sent with a polAssoId that does not exist in the database.
		RESOURCE_URI_STRUCTURE_NOT_FOUND	When resource URI structure is not found.
408	Request Timeout	NA	Supported by Gateway service.
411	Length Required	INCORRECT_LENGTH	Incorrect length.
429	Too Many Requests	NF_CONGESTION_RISK	When there is congestion risk in the NF.
500	Internal Server Error	UNSPECIFIED_NF_FAILURE	When an unexpected error occurs within the service.
503	Service Unavailable	NF_CONGESTION	Supported by Gateway service.

Policy as a producer of Error to SM

Table D-3 Policy as a producer of Error to SM

Error Code	Message	Cause Code	Scenario
400	Bad request	MANDATORY_IE_MISSING	A request is sent without a mandatory IE (SUPI) in the request body.
		UNSPECIFIED_MSG_FAILURE	malformed json object.
		USER_UNKNOWN	A correct request is sent, with a SUPI that is not registered in the UDR.We activate the validate-user in the UE Policy GUI.
		ERROR_INITIAL_PARAMETERS	A request is sent with an error in one or more of its parameters
		PENDING_TRANSACTION	This error shall be used when the PendingTransaction feature is supported and the AMF receives an incomingrequest on a policy association while it has an ongoingtransaction on the same policy association.
401	Unauthorized	With header "WWW-Authenticate"	Supported by Gateway service.
403	Forbidden	MODIFICATION_NOT_ALLOWED	Supported by Gateway service.
		POLICY_CONTEXT_DENIED	When an error occurs in PRE service.

Table D-3 (Cont.) Policy as a producer of Error to SM

		LATE_OVERLAPPING_REQUEST	The request is rejected because it collides with an existing SM context or PDU session context with a more recent origination timestamp.
404	Not Found	SUBSCRIPTION_NOT_FOUND	When UE-Delete is sent with a polAssoId that does not exist in the database.
		RESOURCE_URI_STRUCTURE_NOT_FOUND	This error is being handled by IGW, never reachesthe core service, send 404 Not Found withoud cause.
408	Request Timeout	NA	Supported by Gateway service.
411	Length Required	INCORRECT_LENGTH	The error cause will be changed to ERROR_REQUEST_PARAMETERS.
429	Too Many Requests	NF_CONGESTION_RISK	This will be supported with IGW congestion control and load shedding profiles.
500	Internal Server Error	UNSPECIFIED_NF_FAILURE	The request is rejected due to unspecified reason at the NF.
503	Service Unavailable	NF_CONGESTION	Supported by Gateway service.

Policy as a producer of Error to UE

Table D-4 Policy as a producer of Error to UE

Error Code	Message	Cause Code	Scenario
400	Bad request	MANDATORY_IE_INCORRECT	The SUPI was incorrectly written into the request body and sent to Ingress.
		MANDATORY_IE_MISSING	A request is sent without a mandatory IE (SUPI) in the request body,the cases that are related to this cause will be handled asERROR_REQUEST_PARAMETERS.
		UNSPECIFIED_MSG_FAILURE	When there is malformed JSON object.
		USER_UNKNOWN	A correct request is sent, with a SUPI that is not registered in the UDR.We activate the validate-user in the UE Policy GUI.

Table D-4 (Cont.) Policy as a producer of Error to UE

		ERROR_REQUEST_PARAMETERS	The HTTP request is rejected because the set ofinformation needed by the PCF for UE Policy selection isincomplete or erroneous or not available for the decision tobe made.
		PENDING_TRANSACTION	This error shall be used when the PendingTransactionfeature is supported and the AMF receives an incomingrequest on apolicy association while it has an ongoingtransaction on the same policy association.
401	Unauthorized	With header "WWW-Authenticate"	Supported by Gateway service.
403	Forbidden	MODIFICATION_NOT_ALLOWED	Supported by Gateway service.
404	Not Found	SUBSCRIPTION_NOT_FOUND	When UE-Delete is sent with a polAssoId that does not exist in the database.
		RESOURCE_URI_STRUCTURE_NOT_FOUND	This error is being handled by IGW, never reaches the core service, send 404 Not Found without cause.
408	Request Timeout	NA	Request gets timed out.
411	Length Required	INCORRECT_LENGTH	The error cause will be changed to ERROR_REQUEST_PARAMETERS.
429	Too Many Requests	NF_CONGESTION_RISK	This will be supported with IGW congestion controland load shedding profiles.
500	Internal Server Error	UNSPECIFIED_NF_FAILURE	The request is rejected due to unspecified reason at the NF.
503	Service Unavailable	NF_CONGESTION	Supported by Gateway service.

HTTP Error Codes supported by Policy for CHF and UDR Interfaces

Table D-5 Error codes and cause supported by Policy for CHF Interface

Error Code	Message	Cause Code	Description	Actions
400	Bad Request	INVALID_API	The HTTP request contains an unsupported API name or API version in the URI.	• Session Retry using connector configuration with PUT and POST requests. • After the session retry is exhausted, final error code for PUT or POST failure will reach
		INVALID_MSG_FORMAT	The HTTP request has an invalid format.	

Table D-5 (Cont.) Error codes and cause supported by Policy for CHF Interface

Error Code	Message	Cause Code	Description	Actions
		INVALID_QUERY_PARAM	The HTTP request contains an unsupported query parameter in the URI.	core service for Policy evaluation.
		MANDATORY_QUERY_PARAM_INCORRECT	A mandatory query parameter, or aconditional query parameter but mandatoryrequired, for an HTTP method was received in the URI with semantically incorrect value.	
		OPTIONAL_QUERY_PARAM_INCORRECT	A mandatory query parameter, or aconditional query parameter but mandatoryrequired, for an HTTP method was received in the URI with semantically incorrect value.	
		MANDATORY_QUERY_PARAM_MISSING	Query parameter which is defined asmandatory, or as conditional but mandatoryrequired, for an HTTP method is not included in the URI of the request.	
		MANDATORY_IE_INCORRECT	A mandatory IE or conditional IE butmandatory required, for an HTTP methodwas received with a semantically incorrect.	
		OPTIONAL_IE_INCORRECT		
		MANDATORY_IE_MISSING	A mandatory query parameter is missing in the HTTP request.	
		USER_UNKNOWN	The HTTP request is rejected because PCF received an unknown user.	
		NO_AVAILABLE_POLICY_COUNTERS	Policy counters are not available.	
		UNKNOWN_POLICY_COUNTERS	Policy counters are unknown.	

Table D-5 (Cont.) Error codes and cause supported by Policy for CHF Interface

Error Code	Message	Cause Code	Description	Actions
401	Unauthorized	WWW-Authenticate	Unauthorized with Header "WWW-Authenticate"	• Post Retry with POST request (Session retry) • Put retry with POST (Revalidation/Reset context) • If retry is exhausted, then terminate the transaction.
403	Forbidden	Modification_Not_Allowed	Request is forbidden.	• Retry with POST request • If retry is exhausted, delete the session
404	Not Found	SUBSCRIPTION_NOT_FOUND	The request for modification or deletion of a subscription is rejected because the subscription is not found in the NF.	• Retry with POST request • If retry is exhausted, delete the session
		RESOURCE_URI_STRUCTURE_NOT_FOUND	Resource URI structure is not found.	• Post Retry with POST request (Session retry) • Put retry with POST (Revalidation/Reset context) • If retry is exhausted, then terminate the transaction.
405	Method Not Allowed	NA	The HTTP request is rejected as the performed operation is not allowed.	• Post Retry with POST request (Session retry) • Put retry with POST (Revalidation/Reset context) • If retry is exhausted, then terminate the transaction.
406	Not acceptable	NA	The request is not acceptable.	
409	Conflict	NA	The request is rejected due to schema errors and conflicts in versions.	
410	Gone	NA	This requested resource has been permanently deleted.	
411	Length Required	NA	The HTTP request is rejected due to incorrect value of a Content-length header field.	
412	Precondition Failed	NA	The request is rejected due to incorrect conditions in GET request.	
413	Payload Too Large	NA	The payload is larger than the limit.	
414	URI Too Long	NA	The request URI is longer than the limit.	

Table D-5 (Cont.) Error codes and cause supported by Policy for CHF Interface

Error Code	Message	Cause Code	Description	Actions
415	Unsupported Media Type	NA	The HTTP request contains unsupported payload payload type.	
429	Too many requests	NA	The request is rejected due to excessive traffic, which if continued over time, may lead to (or may increase) an overload situation.	
500	Internal Server Error	NA	The request is rejected due to generic error condition in the NF.	
501	Not Implemented	NA	The service operation cannot be implemented due to invalid resource URI to be monitored.	
503	Service Unavailable	NA	The NF experiences congestion and performs overload control, which does not allow the request to be processed.	
504	Gateway Timeout	NA	This error code is generated in case of timeout due to inactivity.	

Error codes and cause supported by Policy for UDR Interface.

Error codes and cause supported by Policy for UDR Interface:

Error Code	Message	Cause Code	Description	Action
400	Bad Request	INVALID_API	The HTTP request contains an unsupported API name or API version in the URI.	• Session Retry using connector configuration with PUT and POST requests. • If retry is exhausted, terminate the transaction.
		INVALID_MSG_FORMAT	The HTTP request has an invalid format.	
		INVALID_QUERY_PARAM	The HTTP request contains an unsupported query parameter in the URI.	
		MANDATORY_QUERY_PARAM_INCORRECT	A mandatory query parameter, or aconditional query parameter but mandatoryrequired, for an HTTP method was received in the URI with semantically incorrect value.	

Error Code	Message	Cause Code	Description	Action
		OPTIONAL_QUERY_PARAM_INCORRECT	A mandatory query parameter, or aconditional query parameter but mandatoryrequired, for an HTTP method was received in the URI with semantically incorrect value.	
		MANDATORY_QUERY_PARAM_MISSING	Query parameter which is defined asmandatory, or as conditional but mandatoryrequired, for an HTTP method is not included in the URI of the request.	
		MANDATORY_IE_INCORRECT	A mandatory IE or conditional IE butmandatory required, for an HTTP methodwas received with a semantically incorrect.	
		OPTIONAL_IE_INCORRECT		
		MANDATORY_IE_MISSING	A mandatory query parameter is missing in theHTTP request.	
		USER_UNKNOWN	The HTTP request is rejected because PCF received an unknown user.	
401	Unauthorized	WWW-Authenticate	Unauthorized with Header "WWW-Authenticate"	• Post Retry with POST request (Session retry) • Put retry with POST (Revalidation/Reset context) • If retry is exhausted, terminate the transaction
403	Forbidden	Modification_Not_Allowed	Request is forbidden.	Retry with POST request
404	Not Found	SUBSCRIPTION_NOT_FOUND	The request for modification or deletion of a subscription is rejected because the subscription is not found in the NF.	• POST Retry with POST request (Session retry) • Put retry with POST (Revalidation/Reset context) • If retry is exhausted, delete the subscription

Error Code	Message	Cause Code	Description	Action
		RESOURCE_URI_STRUCTURE_NOT_FOUND	Resource URI structure is not found.	• Retry with POST request • If retry is exhausted, terminate the transaction
		USER_UNKNOWN	User is unknown.	• Retry with POST request • If retry is exhausted, terminate the transaction
405	Method Not Allowed	NA	The HTTP request is rejected as the performed operation is not allowed.	• Retry with POST request • If retry is exhausted, terminate the transaction
406	Not acceptable	NA	The request is not acceptable.	• Retry with POST request • If retry is exhausted, terminate the transaction
409	Conflict	NA	The request is rejected due to schema errors and conflicts in versions.	
410	Gone	NA	This requested resource has been permanently deleted.	
411	Length Required	NA	The HTTP request is rejected due to incorrect value of a Content-length header field.	
412	Precondition Failed	NA	The request is rejected due to incorrect conditions in GET request.	
413	Payload Too Large	NA	The payload is larger than the limit.	
414	URI Too Long	NA	The request URI is longer than the limit.	
415	Unsupported Media Type	NA	The HTTP request contains unsupported payload payload type.	
429	Too many requests	NA	The request is rejected due to excessive traffic, which if continued over time, may lead to (or may increase) an overload situation.	NA
500	Internal Server Error	NA	The request is rejected due to generic error condition in the NF.	• Retry with POST request • If retry is exhausted, terminate the transaction
501	Not Implemented	NA	The service operation cannot be implemented due to invalid resource URI to be monitored.	

Error Code	Message	Cause Code	Description	Action
503	Service Unavailable	NA	The NF experiences congestion and performs overload control, which does not allow the request to be processed.	
504	Gateway Timeout	NA	This error code is generated in case of timeout due to inactivity.	

Policy as a producer of Error for Notification for CHF Interface

Table D-6 Policy as a producer of Error for Notification for CHF Interface

Error Code	Message	Cause Code	Scenario
400	Bad Request	MANDATORY_IE_MISSING	When SUPI is not present in the request body for update or terminate notification.
400	Bad Request	INVALID_MESSAGE_FORMAT	When the request body does not adhere to update or terminate notification structure.
400	Bad Request	invalid_API	The HTTP request contains an unsupported API name or API version in the URI.
404	Not Found	RESOURCE_URI_STRUCTURE_NOT_FOUND	The notification URI is incorrect.
404	Not Found	SUBSCRIPTION_NOT_FOUND	When the spendinglimit resource is not found in DB.
504	Gateway Timeout	GATEWAY_TIMEOUT	This error code is generated in case of timeout due to inactivity.

Policy as a producer of Error for Notification for UDR Interface

Table D-7 Policy as a producer of Error for Notification for UDR Interface

Error Code	Message	Cause Code	Scenario
400	Bad Request	INVALID_MESSAGE_FORMAT	When the request body does not adhere to update or terminate notification structure.
404	Not Found	SUBSCRIPTION_NOT_FOUND	When the PolicyDataChangeNotification resource is not found in DB.
501	Not Implemented	UNSUPPORTED_MONITORED_URI	When the PolicyDataChangeNotification received have resource other than smPolicyData,operatorSpecificData, amPolicyData, and UePolicySet.

HTTP Error Codes supported by Policy for NRF Interface

HTTP Error codes supported by Policy for NRF Interface:

Error Code	Message	Description	Action
400	Bad Request	The Request contains invalid values.	**Autonomous mode:** The NF retries primary NRF configurable N times, if fails NF retries secondary NRF configurable M times. NF continues switching back and forth between primary and secondary NRF until success registration. The time interval between one attempt and another is configurable. **On demand mode:** NF retries non primary NRF, if Primary NRF fails. If non primary NRF fails, NF continues with call flow and shall not be able to deregister in NRF.
403	Forbidden	Request is forbidden.	
404	Not Found	The request for modification or deletion of a subscription is rejected because the subscription is not found in the NF.	
405	Method Not Allowed	The request method is not GET, PUT, PATCH or DELETE.	
411	Length Required	The request is rejected due to incorrect value of a Content-length header field, or the header doesn't exist.	
500	Internal Server Error	Internal error occurred in NRF.	
503	Service Unavailable	Rejecting traffic for overload control.	

Policy as a producer of Error for Notification for NRF Interface

Error Code	Message	Cause Code	Scenario
400	Bad Request	INVALID_MSG_FORMAT	NRF terminates the notification transaction.
		MANDATORY_IE_INCORRECT	
		OPTIONAL_IE_INCORRECT	
		MANDATORY_IE_MISSING	
		UNSPECIFIED_MSG_FAILURE	
429	Too Many Requests	NF_CONGESTION_RISK	
500	Internal Server Error	SYSTEM_FAILURE	
503	Service Unavailable	NA	

E
Error Code Dictionary

Table E-1 Error Code Dictionary for SM

App Error ID	Description	Cause	Action
EC-OPCF-SM-INTRNL-EI-05-02-500-00039-05-01	SM Create request from SMF to PCF failed at SM Service **HTTP Status:** 500 - INTERNAL_SERVER_ERROR **Cause:** UNSPECIFIED_NF_FAILURE **Error Category:** Internal (INTRNL)	The request is rejected due to unspecified reasons at the NF.	
EC-OPCF-SM-REQVLD-EI-05-02-400-00048-05-01	SM Create request from SMF to PCF failed at SM Service **HTTP Status:** 400 - BAD_REQUEST **Cause:**ERROR_INITIAL_PARAMETERS **Error Category:** Request validation (REQVLD)	SM Create requests have empty or invalid mandatory request elements	Send create request with all required and valid parameters
EC-OPCF-SM-REQVLD-EI-05-02-400-00010-05-01	SM Create request from SMF to PCF failed SM Service **HTTP Status:** 400 - BAD_REQUEST **Cause:** MANDATORY_IE_MISSING **Error Category:** Request validation (REQVLD)	SM Create request does not have mandatory elements	Send create request with all required parameters
EC-OPCF-SM-REQVLD-EI-05-02-400-00011-05-01	SM Create request from SMF to PCF failed at SM Service **HTTP Status:** 400 - BAD_REQUEST **Cause:** UNSPECIFIED_MSG_FAILURE **Error Category:** Request validation (REQVLD)	SM Create request has an invalid JSON body	Send create request with valid JSON

Table E-1 (Cont.) Error Code Dictionary for SM

App Error ID	Description	Cause	Action
EC-OPCF-SM-REQVLD-EI-05-02-411-00046-05-01	SM Create request from SMF to PCF failed at SM Service **HTTP Status:** 411 - LENGTH_REQUIRED **Cause:** INCORRECT_LENGTH **Error Category:** Request validation (REQVLD)		Send create request with valid JSON and content-length
EC-OPCF-SM-REQVLD-EI-05-02-400-00001-05-01	SM Create request from SMF to PCF failed at SM Service **HTTP Status:** 400 - BAD_REQUEST **Cause:** USER_UNKNOWN **Error Category:** Request validation (REQVLD)	SM Create request has SUPI not registered with UDR and validate-user configuration is enabled in SM Policy GUI.	send request with a SUPI that is registered in the UDR or disable validate-user in the SM
EC-OPCF-SM-SIG-EI-05-02-403-00022-05-01	SM Create request from SMF to PCF failed at SM Service **HTTP Status:** 403 - FORBIDDEN **Cause:** POLICY_CONTEXT_DENIED **Error Category:** Signaling operations errors (SIG)	PCF-SM Policy evaluation returns mandatory action "Reject"SM Create request has an invalid content length for the request body	check policy created, the criteria not matching for the accepting policy. correct policy/send correct request to evaluate it to "Accept"
EC-OPCF-SM-SIG-EI-05-04-404-00025-05-01	SM Delete request from SMF to PCF failed at SM Service **HTTP Status:** 404 - NOT_FOUND **Cause:** SUBSCRIPTION_NOT_FOUND **Error Category:** Signaling operations errors (SIG)	SMF sends a Delete request with an association ID which is not associated with any existing SM session.	send a Delete request with a valid association ID which is associated with the existing SM session.

Table E-1 (Cont.) Error Code Dictionary for SM

App Error ID	Description	Cause	Action
EC-OPCF-SM-SIG-EI-05-02-403-00023-05-01	SM Create request from SMF to PCF failed at SM Service **HTTP Status:** 403 - FORBIDDEN **Cause:** LATE_OVERLAPPING_REQUEST **Error Category:** Signaling operations errors (SIG)	On SMF create Collision detected with older timestamp, request rejected	
EC-OPCF-SM-SIG-EI-05-03-400-00019-05-01	SM Update request from SMF to PCF failed at SM Service **HTTP Status:** 400 - BAD_REQUEST **Cause:** PENDING_TRANSACTION **Error Category:** Signaling operations errors (SIG)	The error occurs when concurrent request arrives and Lock is already acquired by ongoing transaction in SM.	

Table E-2 Error Code Dictionary for AM

App Error Id	Description	Cause	Action
EC-OPCF-AM-REQVLD-EI-05-02-400-00010-06-01	The AM-Create request from AMF to PCF failed at AM Service with HttpStatus - 400 BAD_REQUEST Cause - MANDATORY_IE_MISSING Error Category**:** Request Validation (REQVLD)	A mandatory Information element is missing in request.	Send the Create request with required parameters in the request body.
EC-OPCF-AM-REQVLD-EI-05-02-400-00011-06-01	The AM-Create request from AMF to PCF failed at AM Service with HttpStatus - 400 BAD_REQUEST Cause - UNSPECIFIED_MSG_FAILURE Error Category**:** Request Validation (REQVLD)	The request is rejected due to unspecified client error.	Make sure the request body is in right Json format.

Table E-2 (Cont.) Error Code Dictionary for AM

App Error Id	Description	Cause	Action
EC-OPCF-AM-REQVLD-EI-05-03-400-00011-06-01	The AM-Update request from AMF to PCF failed at AM Service with HttpStatus - 400 BAD_REQUEST Cause - UNSPECIFIED_MSG_FAILURE Error Category: Request Validation (REQVLD)	The request is rejected due to unspecified client error.	Make sure the request body is in right Json format.
EC-OPCF-AM-REQVLD-EI-05-02-400-00048-06-01	The AM-Create request from AMF to PCF failed at AM Service with HttpStatus - 400 BAD_REQUEST Cause - ERROR_INITIAL_PARAMETERS Error Category: Request Validation (REQVLD)	A request is sent with an error in one or more of its parameters.	Send the Create request with right values for the mandatory parameters.
EC-OPCF-AM-REQVLD-EI-05-02-411-00046-06-01	The AM-Create request from AMF to PCF failed at AM Service with HttpStatus - 411 Length Required Cause - INCORRECT_LENGTH Error Category: Request Validation (REQVLD)	The request is rejected because the length of a provided value does not meet the expected or required criteria.	Make sure that the length of the input matches the required specifications.
EC-OPCF-AM-REQVLD-EI-05-03-411-00046-06-01	The AM-Update request from AMF to PCF failed at AM Service with HttpStatus - 411 Length Required Cause - INCORRECT_LENGTH Error Category: Request Validation (REQVLD)	The request is rejected because the length of a provided value does not meet the expected or required criteria.	Make sure that the length of the input matches the required specifications.
EC-OPCF-AM-REQVLD-EI-05-02-400-00018-06-01	The AM-Create request from AMF to PCF failed at AM Service with HttpStatus - 400 BAD_REQUEST Cause - ERROR_REQUEST_PARAMETERS Error Category: Request Validation (REQVLD)	The Create request is rejected because the information set required by the PCF for AM Policy selection is incomplete, preventing a decision from being made.	Make sure the required parameters are in expected format.

Table E-2 (Cont.) Error Code Dictionary for AM

App Error Id	Description	Cause	Action
EC-OPCF-AM-REQVLD-EI-05-02-400-00001-06-01	The AM-Create request from AMF to PCF failed at AM Service with HttpStatus - 400 BAD_REQUEST Cause - USER_UNKNOWN Error Category: Request Validation (REQVLD)	When validate user is enabled and Create request is sent for a SUPI hat is not registered in the UDR.	Send the Create request with right/registered Supi.
EC-OPCF-AM-SIG-EI-05-03-404-00025-06-01	The AM-Update request from AMF to PCF failed at AM Service with HttpStatus - 404 Not Found Cause - SUBSCRIPTION_NOT_FOUND Error Category: Signalling (SIG)	When AM-Update is sent with a polAssoId that is not associated with any existing AM session.	Send the Update request with right/ registered Supi.
EC-OPCF-AM-SIG-EI-05-04-404-00025-06-01	The AM-Terminate request from AMF to PCF failed at AM Service with HttpStatus - 404 Not Found Cause - SUBSCRIPTION_NOT_FOUND Error Category: Signalling (SIG)	When AM-Delete is sent with a polAssoId that is not associated with any existing AM session.	Send the Delete request with right/registered polAssoId.
EC-OPCF-AM-INTRNL-EI-05-02-500-00039-06-01	The AM-Create request from AMF to PCF failed at AM Service with HttpStatus - 500 Internal Server Error Cause - UNSPECIFIED_NF_FAILURE Error Category: Internal (INTRNL)	The request is rejected due to unspecified reason at the NF.	

Table E-2 (Cont.) Error Code Dictionary for AM

App Error Id	Description	Cause	Action
EC-OPCF-AM-INTRNL-EI-05-03-500-00039-06-01	The AM-Update request from AMF to PCF failed at AM Service with HttpStatus - 500 Internal Server Error Cause - UNSPECIFIED_NF_FAILURE Error Category: Internal (INTRNL)	The request is rejected due to unspecified reason at the NF.	
EC-OPCF-AM-INTRNL-EI-05-04-500-00039-06-01	The AM-Terminate request from AMF to PCF failed at AM Service with HttpStatus - 500 Internal Server Error Cause - UNSPECIFIED_NF_FAILURE Error Category: Internal (INTRNL)	The request is rejected due to unspecified reason at the NF.	
EC-OPCF-AM-SIG-EI-05-02-403-00022-06-01	The AM-Create request from AMF to PCF failed at AM Service with HttpStatus - 403 Forbidden Cause - POLICY_CONTEXT_DENIED Error Category: Signalling (SIG)	AM Create request is rejected by PRE	Check and update PRE policy accordingly
EC-OPCF-AM-SIG-EI-05-03-403-00022-06-01	The AM-Update request from AMF to PCF failed at AM Service with HttpStatus - 403 Forbidden Cause - POLICY_CONTEXT_DENIED Error Category: Signalling (SIG)	AM Update request is rejected by PRE	Check and update PRE policy accordingly

Table E-2 (Cont.) Error Code Dictionary for AM

App Error Id	Description	Cause	Action
EC-OPCF-AM-SIG-EI-05-04-403-00022-06-01	The AM-Terminate request from AMF to PCF failed at AM Service with HttpStatus - 403 Forbidden Cause - POLICY_CONTEXT_DENIED Error Category: Signalling (SIG)	AM Terminate request is rejected by PRE	Check and update PRE policy accordingly
EC-OPCF-AM-SIG-EI-05-02-403-00023-06-01	The AM-Create request from AMF to PCF failed at AM Service with HttpStatus - 403 Forbidden Cause - LATE_OVERLAPPING_REQUEST Error Category: Signalling (SIG)	The request is rejected because it collides with an existing AM context or PDU session context with a more recent origination timestamp.	
EC-OPCF-AM-SIG-EI-05-03-400-00019-06-01	The AM-Update request from AMF to PCF failed at AM Service with HttpStatus - 400 Bad Request Cause - PENDING_TRANSACTION Error Category: Signalling (SIG)	This error shall be used when the PendingTransaction feature is supported and the PCF receives an incoming request on a policy association while it has an ongoing transaction on the same policy association.	

Table E-3 Error Code Dictionary for UE

App Error Id	Description	Cause	Action
EC-OPCF-UE-REQVLD-EI-05-02-400-00010-06-01	UE Create request from AMF to PCF failed at UE Service. **HTTP Status:** 400 - BAD_REQUEST **Cause:** MANDATORY_IE_MISSING **Error Category:** Request Validation (REQVLD)	UE Create request does not have mandatory information elements	Send create request with all required parameters

Table E-3 (Cont.) Error Code Dictionary for UE

App Error Id	Description	Cause	Action
EC-OPCF-UE-REQVLD-EI-05-02-400-00011-06-01	UE Create request from AMF to PCF failed at UE Service. **HTTP Status**: 400 - BAD_REQUEST **Cause:** UNSPECIFIED_MSG_FAILURE **Error Category:** Request Validation (REQVLD)	UE Create request have invalid JSON body	Send create request with valid JSON body
EC-OPCF-UE-REQVLD-EI-05-02-400-00001-06-01	UE Create request from AMF to PCF failed at UE Service. **HTTP Status:** 400 - BAD_REQUEST **Cause:** USER_UNKNOWN **Error Category:** Request Validation (REQVLD)	UE Create request have unregistered SUPI. **Configuration:** Validate user flag is enabled	Send create request with a registered SUPI
EC-OPCF-UE-REQVLD-EI-05-02-400-00018-06-01	UE Create request from AMF to PCF failed at UE Service. **HTTP Status:** 400 - BAD_REQUEST **Cause:** ERROR_REQUEST_PARAMETERS **Error Category:** Request Validation (REQVLD)	UE Create request have empty mandatory information elements	Send create request with all required and valid parameters
EC-OPCF-UE-SIG-EI-05-02-400-00019-06-01	UE Create request from AMF to PCF failed at UE Service. **HTTP Status:** 400 - BAD_REQUEST **Cause:** PENDING_TRANSACTION **Error Category:** **Signaling (SIG)**	UE Create request cannot be processed because of the pending transaction of same policy association in request.	

Table E-3 (Cont.) Error Code Dictionary for UE

App Error Id	Description	Cause	Action
EC-OPCF-UE-SIG-EI-05-02-403-00022-06-01	UE Create request from AMF to PCF failed at UE Service. **HTTP Status:** 403 - FORBIDDEN **Cause:** POLICY_CONTEXT_DENIED **Error Category:** Signaling (SIG)	UE Create request rejected by PRE	Check and update PRE policy accordingly if needed
EC-OPCF-UE-SIG-EI-05-02-403-00023-06-01	UE Create request from AMF to PCF failed at UE Service. **HTTP Status:** 403 - FORBIDDEN **Cause:** LATE_OVERLAPPING_REQUEST **Error Category:** Signaling (SIG)	UE Create request rejected because it collides with an existing UE context or PDU session context with more recent origination timestamp.	
EC-OPCF-UE-REQVLD-EI-05-02-411-00046-06-01	UE Create request from AMF to PCF failed at UE Service. **HTTP Status:** 411 - LENGTH REQUIRED **Cause:** INCORRECT_LENGTH **Error Category:** Request Validation (REQVLD)	Server (*i.e. PCF, specifically UE Services*) rejects the request with incorrect Content-Length header	Check if Content-Length header is present in request headers and has a valid value which matches to the request body size.
EC-OPCF-UE-INTRNL-EI-05-02-500-00039-06-01	UE Create request from AMF to PCF failed at UE Service. **HTTP Status:** 500 INTERNAL SERVER ERROR **Cause:** UNSPECIFIED_NF_FAILURE **Error Category:** Internal (INTRNL)	UE Create request failed with unexpected internal service error.	

Table E-3 (Cont.) Error Code Dictionary for UE

App Error Id	Description	Cause	Action
EC-OPCF-UE-REQVLD-EI-05-03-400-00011-06-01	UE Update request from AMF to PCF failed at UE Service. **HTTP Status**: 400 - BAD_REQUEST **Cause:** UNSPECIFIED_MSG_FAILURE **Error Category:** Request Validation (REQVLD)	UE Update request have invalid JSON body	Send update request with valid JSON body
EC-OPCF-UE-SIG-EI-05-03-403-00022-06-01	UE Update request from AMF to PCF failed at UE Service. **HTTP Status:** 403 - FORBIDDEN **Cause:** POLICY_CONTEXT_DENIED **Error Category:** Signaling (SIG)	UE Update request rejected by PRE	Check and update PRE policy accordingly if needed
EC-OPCF-UE-SIG-EI-05-03-403-00023-06-01	UE Update request from AMF to PCF failed at UE Service. **HTTP Status:** 403 - FORBIDDEN **Cause:** LATE_OVERLAPPING_REQUEST **Error Category:** Signaling (SIG)	UE Update request rejected because it collides with an existing UE context or PDU session context with more recent origination timestamp.	
EC-OPCF-UE-SIG-EI-05-03-404-00025-06-01	UE Update request from AMF to PCF failed at UE Service. **HTTP Status:** 404 - NOT FOUND **Cause:** SUBSCRIPTION_NOT_FOUND **Error Category:** Signaling (SIG)	UE Update request failed because of unregistered association ID in URL.	Send update request with a registered association ID
EC-OPCF-UE-REQVLD-EI-05-03-411-00046-06-01	UE Update request from AMF to PCF failed at UE Service. **HTTP Status:** 411 - LENGTH REQUIRED **Cause:** INCORRECT_LENGTH **Error Category:** Request Validation (REQVLD)	Server (*i.e. PCF, specifically UE Services*) rejects the request with incorrect Content-Length header	Check if Content-Length header is present in request headers and has a valid value which matches to the request body size.

Table E-3 (Cont.) Error Code Dictionary for UE

App Error Id	Description	Cause	Action
EC-OPCF-UE-INTRNL-EI-05-03-500-00039-06-01	UE Update request from AMF to PCF failed at UE Service. **HTTP Status:** 500 INTERNAL SERVER ERROR **Cause:** UNSPECIFIED_NF_FAILURE **Error Category:** Internal (INTRNL)	UE Update request failed with unexpected internal service error.	
EC-OPCF-UE-SIG-EI-05-04-404-00025-06-01	UE Delete request from AMF to PCF failed at UE Service. **HTTP Status:** 404 - NOT FOUND **Cause:** SUBSCRIPTION_NOT_FOUND **Error Category:** Signaling (SIG)	UE Delete request failed because of unregistered association ID in URL.	Send update request with a registered association ID
EC-OPCF-UE-INTRNL-EI-05-04-500-00039-06-01	UE Delete request from AMF to PCF failed at UE Service. **HTTP Status:** 500 INTERNAL SERVER ERROR **Cause:** UNSPECIFIED_NF_FAILURE **Error Category:** Internal (INTRNL)	UE Update request failed with unexpected internal service error.	

Table E-4 Error Code Dictionary for UDR Connector

App Error Id	Description	Cause	Action
EC-OPCF-UDR_C-REQVLD-EI-05-08-400-00003-03-01	The UDR notification request from UDR to PCF failed at UDR-Connector with httpStatus- 400 BAD_REQUEST cause-INVALID_MSG_FORMAT Note: Applies for UM notification as well	1. Incorrect JSON format 2. Empty Notification Body i.e., []	1. Verify UDR to send the content with right JSON format 2. Verify UDR to send the notification with notification data

Table E-4 (Cont.) Error Code Dictionary for UDR Connector

App Error Id	Description	Cause	Action
EC-OPCF-UDR_C-REQVLD-EI-05-08-400-00011-03-01	The UDR notification request from UDR to PCF failed at UDR-Connector with httpStatus- 400 BAD_REQUEST cause-UNSPECIFIED_MSG_FAILURE Note: Applies for UM notification as well	1. Null notification body i.e., no content in the body	1. Verify UDR to send the notification with content
EC-OPCF-UDR_C-REQVLD-EI-05-08-501-00043-03-01	The UDR notification request from UDR to PCF failed at UDR-Connector with httpStatus- 501 Not Implemented cause-UNSUPPORTED_MONITORED_URI Note: Applies for UM notification as well	1. When notification body doesn't contain atleast one of the data source notification data among the below a. smPolicyData b. amPolicyData c. uePolicySet d. delResources	1. Verify UDR to send the notification with atleast one of the listed dataSource information
EC-OPCF-UDR_C-EI-05-08-404-00025-03-01	The UDR notification request from UDR to PCF failed at UDR-Connector with httpStatus- 404 Not Found cause-SUBSCRIPTION_NOT_FOUND Note: Applies for UM notification as well	1. When the notification received user session is not present in Policy-ds 2. Session for the user is found but subscription information is missing.	

Table E-5 Error Code Dictionary for CHF Connector

App Error Id	Description	Cause	Action
EC-OPCF-CHF_C-REQVLD-EI-05-08-400-00003-04-01	The NOTIFY-POST request from CHF to PCF failed with status code 400 and cause 'INVALID_MSG_FORMAT'	The request body JSON need to follow a certain format	Send the POST request in the correct format.
EC-OPCF-CHF_C-REQVLD-EI-05-08-400-00010-04-01	The NOTIFY-POST request from CHF to PCF failed with status code 400 and cause 'MANDATORY_IE_MISSING'	SUPI is a mandatory parameter for the POST request	Send the POST request with the SUPI information present in the body.

Table E-5 (Cont.) Error Code Dictionary for CHF Connector

App Error Id	Description	Cause	Action
EC-OPCF-CHF_C-REQVLD-EI-05-08-400-00011-04-01	The NOTIFY-POST request from CHF to PCF failed with status code 400 and cause 'UNSPECIFIED_MSG_FAILURE'	The request body JSON needs to follow a certain format	Send the POST request in the correct format.
EC-OPCF-CHF_C-SIG-EI-05-08-404-00025-04-01	The NOTIFY-POST request from CHF to PCF failed with status code 404 and cause 'SUBSCRIPTION_NOT_FOUND'	The user session is not present in PolicyDS or the session for the user is found but the subscription information is missing.	
EC-OPCF-CHF_C-INTRNL-EI-05-08-500-00000-04-01	The NOTIFY-POST request from CHF to PCF failed with status code 500 and cause 'UNKNOWN'	PDS responded back with status code 500	
EC-OPCF-CHF_C-INTRNL-EI-05-08-503-00000-04-01	The NOTIFY-POST request from CHF to PCF failed with status code 503 and cause 'UNKNOWN'	Required service is unavailable	Check whether all services are up and running
EC-OPCF-CHF_C-REQVLD-EI-05-11-400-00003-04-01	The TERMINATE-POST request from CHF to PCF failed with status code 400 and cause 'INVALID_MSG_FORMAT'	The request body JSON need to follow a certain format	Send the POST request in the correct format.
EC-OPCF-CHF_C-REQVLD-EI-05-04-400-00011-04-01	The TERMINATE-POST request from CHF to PCF failed with status code 400 and cause 'UNSPECIFIED_MSG_FAILURE'	The request body JSON needs to follow a certain format	Send the POST request in the correct format.
EC-OPCF-CHF_C-SIG-EI-05-08-404-00025-04-01	The TERMINATE-POST request from CHF to PCF failed with status code 404 and cause 'SUBSCRIPTION_NOT_FOUND'	The user session is not present in PolicyDS or the session for the user is found but the subscription information is missing.	
EC-OPCF-CHF_C-INTRNL-EI-05-08-500-00000-04-01	The TERMINATE-POST request from CHF to PCF failed with status code 500 and cause 'UNKNOWN'	PDS responded back with status code 500	
EC-OPCF-CHF_C-INTRNL-EI-05-08-503-00000-04-01	The TERMINATE-POST request from CHF to PCF failed with status code 503 and cause 'UNKNOWN'	Required service is unavailable	Check whether all services are up and running

Table E-6 Error Code Dictionary for Binding

App Error Id	Description	Cause	Action
EC-OPCF-BINDING-REQVLD-EI-05-19-500-00011-02-01	BSF Audit request from BSF to PCF failed at Binding Service with HttpStatus - 500 Internal Server Error Cause - UNSPECIFIED_MSG_FAILURE Error Category: Request Validation (REQVLD)	The request is rejected due to unspecified client error.	Make sure the request body is in right Json format.
EC-OPCF-BINDING-REQVLD-EI-05-19-500-00046-02-01	BSF Audit request from BSF to PCF failed at Binding Service with HttpStatus - 500 Internal Server Error Cause - INCORRECT_LENGTH Error Category: Request Validation (REQVLD)	The request has an invalid content length for the request body	Check if Content-Length header is present in request headers and has a valid value which matches to the request body size.
EC-OPCF-BINDING-SIG-EI-05-19-404-00053-02-01	BSF Audit request from BSF to PCF failed at Binding Service with HttpStatus - 404 Not Found Cause - CONTEXT_NOT_FOUND Error Category: Signalling (SIG)	BSF Audit request is sent with a context id that is not associated with any existing Binding session.	Send the request with registered context id.
EC-OPCF-BINDING-INTRNL-EI-05-19-500-00000-02-01	BSF Audit request from BSF to PCF failed at Binding Service with HttpStatus - 500 Internal Server Error Cause - UNKNOWN Error Category: Internal (INTRNL)	The request is rejected due to unknown error.	